From Advanced Digital Signal Processing to Machine Learning

Jun Zhou, *Member, IEEE* and Hen-Geul Yeh, *Senior Member IEEE*

***Abstract—Contribution*: A modular three-credit senior undergraduate or graduate course that includes both advanced digital signal processing (DSP) and machine learning (ML) is introduced. This project-based course is featured at exploiting the connections between the two popular areas. It presents the advantages and limitations of each area to the students with the help of various specially designed course projects.**

***Background:* At most schools, DSP and ML are taught in two separate courses although the two have many similarities in both theorems and applications. It's partially due to the fact that the two areas are largely supported by two research communities in the literature.**

***Intended Outcomes*: Students understand the fundamentals of the DSP and the ML from both theorems and applications perspectives with a clear picture of the advantages and limitations in the state-of-the-art. Set up initial motivations and inspire the students to the future research activities.**

***Application Design*: An approach that concentrated content and project work on the modular domain knowledge was taken. The authors started with the probability and statistics, followed by a series of carefully selected topics from each area that can be combined in a systematic way, from theorems to applications, from algorithms to implementations. An oral presentation session was included to emphasize independent analysis and professional skills development in the student selected project.**

***Findings*: Based on the course survey, the students reported positive feedbacks on the newly structured course. It showed that this course can benefit from the strategic introduction of desirable technology in the curriculum efficiently. From the theorems perspective, students reported the domain knowledge of the two areas. From the applications perspective, students were exposed to the state-of-the-art tools in two disciplines, career preparation, and in-class engagement. A direct assessment showed a promising result.**

***Index Terms*—advanced digital signal processing, machine learning, professional skills, project-based learning**

I. Introduction

THIS section provides a brief review of recent development of the digital signal processing (DSP) and machine learning (ML) areas [1-11]. We highlight the similarities as well as the differences from the perspectives of background, theorems, and the applications [2, 4, 6-8, 10]. For examples, they interconnect in areas like music, voice recognition, adaptive filter design, pattern recognition, robotics, etc. From systems engineering point of view, both DSP and ML work on solving problems and equations in systems level. ML explores a space of solutions in a more "heuristic" way, while DSP uses mathematical tools like transforms, correlations [2], optimization, and analytical approaches. Moreover, we highlight the necessities of combining the two together for better educational purpose as well as better research preparation [7, 9, 11].

Advances in DSP are often fast-paced and innovative. Lately, demand is growing for faster processor architectures to support embedded vision and artificial intelligent (AI) applications [4]. Traditionally, DSP curriculum starts from undergraduate linear time-invariant systems to graduate time-variant adaptive signal processing systems. It grows from a simple 1-dimensional single input, single layer, single output architecture to 2-dimensional multiple inputs, multiple layers, multiple outputs neural network architecture [11]. Without loss of generality and for the sake of discussion, AI and ML are interchangeable in this paper.

AI is currently revolutionizing how we perform modeling, detection, classification, and control complex systems in modern era. These systems are typically nonlinear, dynamic, multi-scale and multi-variable, in space and time. They are high-dimensional with dominant underlying patterns that should be characterized and modeled for prediction and estimation. The substantial increase in research, development, and applications in ML and AI should rapid educational institutions to comprehensively train scientists and engineers with knowhow, understanding, and experience from DSP or other related fields in this exciting area.

In general, advanced undergraduate and beginning graduate students are interested in learning both DSP and ML. These students have backgrounds in linear algebra, differential or difference equations, and scientific computation using MATLAB or Python. However, most undergraduate curriculum in Electrical Engineering has little or no exposure to data-driven methods. Likewise, most undergraduate curriculum in Computer Science has little exposure to digital signal processors or FPGAs with applications to real-time DSP, audio, image, communication, control, and smart grid systems. Our goal in this paper is to provide a broad entry point to connect DSP and ML for both of these groups of students and other related engineers, and scientists. Some authors have suggested that DSP and ML education may start early for smooth learning and better education [1-3].

There are many DSP algorithms used in industrial applications and academic research, for example, convolution, autocorrelation, fast Fourier transform (FFT) [9] and wavelets transforms [12], convolution [2], adaptive filtering via least mean squares (LMS) [5] or recursive least squares (RLS) [5], linear estimators, compressed sensing and gradient descent, to mention a few. Specifically, FFT, LMS, and RLS are applied to and time-variant adaptive signal processing systems. These algorithms cover a wide range from 1-dimensional single input, single layer, single output relationship to 2-dimensional multiple inputs, multiple layers, multiple outputs. However, these algorithms are also applied in ML in a fairly straightforward [2, 5] regardless of what deep neural network architecture used.

Though DSP and ML share quite some common knowledge, such as probability, statistics and linear regression, they are largely supported by two different research communities. For example, IEEE has dedicated societies for DSP and ML, such as signal processing society and computational intelligence society. Though the number of cross-disciplines articles have increased year by year, there are different domain conference and journals between these two areas. As a result, one course is mostly focused on one area usually has little or no coverage of the other. Highlight the divergence of the course topics since decades ago (DSP is likely to cover filtering, communication theory, audio/image processing, etc.; while ML is more focused on feature engineering, neural networks, etc.) In this paper, we present the course structure as depicted in Fig. 1. It is an integration of DSP course and ML course as one single course for senior undergraduate or first year graduate students to meet demands from student's sides.

Moreover, we emphasize Markov Model in statistical course [13]. Design Thinking method [14] is applied in teaching these project-based engineering courses in that it incorporates business-minded concerns, such as attention to the end user and economics of the proposed solution, including cost. This broad approach is desirable to industry and thought to be a promising way forward in bridging the gaps between engineering education and the workplace [14 - 15]. The graduate class size is smaller than the undergraduate for learning efficiency [16].

The course is offered once per year. Teaching effectiveness is a complex construct and includes numerous dimensions, behaviors, skills, and characteristics. Teaching effectiveness includes teacher characteristics that influence students' attitudes and behaviors in the learning environment (e.g., knowledgeable, approachable, interesting, and motivating). As the demonstration, student learning is a desired outcome of teaching effectiveness. Course evaluation from students is the most valuable feedback data to instructors. Student involvement in this course is considered as part of course evaluation from students in classroom. Students comments and suggestions are considered for improving the course materials in next teaching cycle.

This paper includes following sections. The course design is discussed in Section II, where we introduce the course structure and the topics selection based on the domain knowledge. The course implementation in California State University in Long Beach (CSULB) is presented in Section III. We elaborate the course materials in this section, including project assignments, quiz, exam and oral presentation in the course. The student feedback by course survey is given in Section IV. Finally, Section V concludes the paper.

II. Course Design

In this section, we present the proposed course structure, lecture content and project development. Fig. 1 shows the block diagram of course structure. This course includes four topics from DSP and ML areas. They are sigma delta modulation and adaptive filtering in DSP, and linear regression and neural network in ML. These topics cover various aspects of a modern smart system, including signal acquisition, signal conditioning, processing, and signal detection, etc. In addition, they also maximize the interconnection between the DSP and ML. There are two education goals in this course. The first goal is to prepare students of Electrical Engineering and Computer Science with a systematical overview of these two most popular research areas by leveraging their existing knowledge. Students will see both DSP and ML share some common background, while they are most efficient in dealing with different questions in smart systems. The second goal is to let students explore both hardware and software aspects of a smart system. [4-9, 11-12]

The background knowledge of linear algebra, probability and statistics, signals and systems are the prerequisites of this course. Computational programming is utilized throughout the course. Two computational programming software languages are adopted in this course to demonstrate the theorems and the projects, i.e. MATLAB for DSP related topics and Python for ML related topics. We recommend the students to have some programming experience on MATLAB and/or Python. If not, C/C++ experience is required. According to our teaching practices in CSULB, these prerequisites are not an issue to most of the advanced undergraduate and beginning graduate students in engineering school.

The following course structure is based on a semester of 16 weeks, including final exam week. It is flexible to fit into two quarters as well with some expansion on the topics. A brief discussion on this is included in each subsection. We start with a brief review of the DSP and ML, including statistics, linear programming, and spectrum analysis as they are shared fundamentals of DSP and ML. Next, we use sigma delta modulation and adaptive filtering in the DSP area to cover the loop filter optimization. Then, we use linear optimization tools and neural networks to cover the most recent development in the ML area. Note that, these four topics frequently appear in the modern smart systems, which consists of analog to digital conversion, noise filtering and conditioning, cost optimization, detection and classification. In addition, the selected topics also cover implementations and hardware cost so that students have a concept of hardware oriented algorithm design with both software (i.e. algorithms) and hardware (i.e. DSP processors, FPGA, GPUs).

We adopt a combination of quiz, project, examination and oral presentation as evaluation methods. The quiz scans the knowledge understanding. The comprehensive project tests the skills of applying theorems and tools to the complex problems. Oral presentation motivates the students to present the project study in a former manner that is similar to an oral presentation in an academic conference.

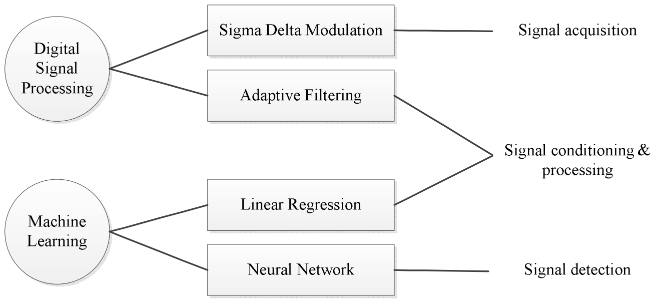

Fig. 1. Block diagram of the proposed course structure.

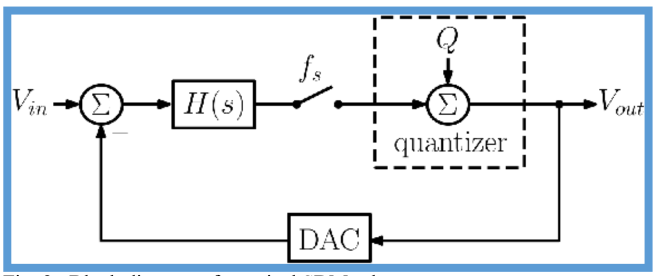

Fig. 2. Block diagram of a typical SDM scheme.

A. Background Review

In the beginning of the course, we provided a brief review of the DSP and the ML topics. The course roadmap is illustrated in Fig. 1, that includes signal acquisition, noise and interference mitigation, optimization, pattern detection and classification. We emphasized the advantages as well as the limitations of the DSP and ML approaches at the lectures. For example, the DSP approach is great to deal with low-dimensional, usually noise corrupted, and unstructured data. Some good DSP examples are, but not limited to, audio and speech processing [2, 11], radar processing [17-18], signal detection and synchronization in communication [4-5, 18]. On the other side, the ML approach is excellent in processing high-dimensional, little noise, and well-structured data. Some good ML examples are, but not limited to, autonomous driving [24], speech and face recognition [25], object detection [12, 20-23]. Given this background, the students can demystify what are DSP and ML and what are they good at.

We introduced MATLAB and Python programming languages and helped students install the software, if they have not done it yet (due to the limited access to the shared computing throughout the 2020 academic year). We used these two languages because MATLAB and Python are largely supported by the DSP and the ML communities. Regarding MATLAB installation, we use a standard installation procedure. Regarding Python installation, it's worth noting that Python has many good distributions, which are different from MATLAB. Coming to this course, it's important to adopt a convenient way to maintain the Python packages and environment between the instructors and the students for evaluation purposes. We adopt Anaconda as the Python distribution due to its simplified package management and deployment [27].

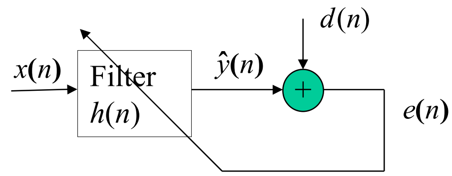

Fig. 3. The block diagram of a basic LMS adaptive filtering scheme.

B. DSP topic 1: Sigma Delta Modulation (SDM)

We select SDM as the first DSP topic based on three facts: (1) it's necessary to convert a signal from analog to digital (ADC) equivalent before any comprehensive algorithms, no matter it's a DSP approach or a ML approach; (2) DSM is one of the most popular analog to digital conversion techniques; (3) DSM itself includes several different DSP concepts, such as filtering, closed-loop system, signal sampling and quantization.

In this topic, students will learn the functionalities of an analog to digital converter in electronic devices. The analog to digital conversion is nonlinear and non-recoverable. The quantization noise Q in Fig. 1 is inevitable during analog to digital conversion. The signal to quantization plus noise ratio (SQNR) in a Nyquist sampling system is shown in Eq. (1)

$$SQNR = 6.02N + 1.76 \ \ (dB) \tag{1}$$

In order to minimize the quantization noise, we can either increase the number of quantization bits or increase the sampling rate. A more efficient way is to employ a loop filter to push the quantization noise to higher frequency spectrum. This is the fundamentals of SDM.

This topic starts with a simple first-order low-pass filter and its spectrum characteristics. Then we analyze the spectrum characteristics in a closed-loop system depending on the signal injection place. It can be either a low-pass filter (LPF) or a high-pass filter (HPF) in a closed-loop system. Coming to the sampling and quantization schemes, we demonstrate that the signal goes through with an equivalent LPF and the noise Q goes through with an equivalent HPF. Fig. 2 illustrates the block diagram of a typical SDM. $H(s)$ is a low pass filter which is also known as loop filter. The switch represents sampling and hold circuitry. It is followed by a quantizer, which converts an analog and continuous signal into digital and discrete format. The digital to analog converter (DAC) on the feedback path provides a matching template to the input.

The goal of this topic is to cover the architecture of a typical signal processing system. Students will learn what is analog to digital conversion, and what the role is in the entire system. It is also a great opportunity to elaborate the hardware consideration in a system design. Student will understand the algorithm and implementation from a system level. Though this topic starts with simple ideas, it can be extended to cover wilder aspects, like closed-loop system analysis, hardware imperfection, etc.

C. DSP topic 2: Adaptive Filtering

Adaptive filtering is chosen as the second DSP topic. It connects to the first DSP topic in the concept of closed-loop system. However, adaptive filtering is a dynamic system by nature (compared to a static system as SDM) and it handles the optimization of one or more parameters simultaneously. It also has connections to some of the ML topics, for example the concept of cost function, gradient-descend solver, etc.

In this topic, students will learn how to build a cost function and how to solve it from the signal processing perspective. Fig. 3 shows the block diagram of a basic adaptive filtering scheme using least mean square (LMS) algorithm.

LMS adaptively finds the optimal weight to minimize the mean square error. With the stochastic gradient approximation, the LMS algorithms are summarized in Table 1. Note that $x(n)$ and $e(n)$ are the input and error sequences with time index n, respectively, $\hat{y}(n)$ is the filter output sequence and $d(n)$ is the desired or reference sequences, μ is a small adaptive constant or step size, N is the length of the filter, $\boldsymbol{h}(n)$ is the filter weighting vector, which is updated recursively.

TABLE I: SUMMARY OF THE LMS ALGORITHMS

I.C. $\mathbf{h}(n) = 0, \quad 0 \le n \le N-1$

1. $t = n$, $\mathbf{h}(n)$ is available

2. Compute $\hat{y}(n) = \sum_{j=0}^{N-1} h(j)x(n-j)$

3. $e(n) = d(n) - \hat{y}(n)$

4. $\mathbf{h}(n+1) = \mathbf{h}(n) + 2\mu e(n)\mathbf{x}(n)$

$$\text{where } \mathbf{h}(n) = \begin{bmatrix} h(0) \\ h(1) \\ : \\ h(N-1) \end{bmatrix}, \quad \mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ : \\ x(n-N+1) \end{bmatrix}$$

5. $n -> n+1$ Go To Step 1.

The goal of this topic is to introduce a denoising technique from the DSP perspective. It will be compared to the ML topics. Actually, there are research activities that adopt ML principles to analyze ECG signals on GitHub. Students can be guided to expand the readings on that.

D. *ML topic 1: Linear Regression*

We select linear regression as the first ML topic based on three facts: (1) it's a fundamental topic in ML that has been studied rigorously; (2) it shares some similarities to the second DSP topic adaptive filtering, such as the cost function, gradient descent solver, etc.; (3) it can be realized by either conventional signal processing approach or neural networks.

In statistics, linear regression is a set of approaches that use linear functions to model the relationship between a scalar response and one or multiple explanatory variables. A simple linear regression involves only one explanatory variable, while multiple linear regression deals with multiple explanatory variables simultaneously. Note that, generalized linear models

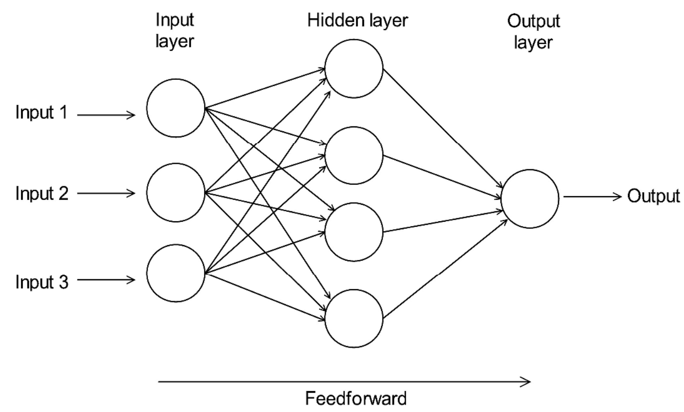

Fig. 4 Illustration of a simplified feedforward artificial neural network.

can learn non-linear mapping from input to output as long as only the linear part of the model learns.

Eqs. (2) and (3) have the same number of adaptive coefficients as the dimensionality of the data plus 1. We can see that fitting the second model is exactly the same problem as fitting the first model once we have replaced the data by the outputs of the basis functions $\phi_i(\mathbf{x}), i = 1, ..., N-1$.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots = \mathbf{w}^T \mathbf{x} \tag{2}$$

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 \phi_1(\mathbf{x}) + w_2 \phi_2(\mathbf{x}) + \ldots = \mathbf{w}^T \Phi(\mathbf{x}) \tag{3}$$

A loss function evaluates the error of a model prediction to the actual measurement. Fitting a model to data is usually be done by searching parameters that minimize a certain loss function. In linear regression, a loss function typically consists of a squared error term. A least squares estimator is

$$\begin{aligned} y &= \mathbf{w}^T \mathbf{x} \\ error &= \sum_n \left(t_n - \mathbf{w}^T \mathbf{x}_n\right)^2 \\ \mathbf{w}^* &= \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{t} \end{aligned} \tag{4}$$

The goal of this project is to get students familiar with linear regression tasks and neural networks solution via Tensorflow.

E. *ML topic 2: Neural Networks*

We select neural networks as the second ML topic. It is an emerging research area that has drawn great attention from both computer science and electrical engineering. Modern neural networks are developed to solve AI problems. It has demonstrated great precision, robustness and flexibility in many applications, such as autonomous driving, audio and image enhancement, recognition and classification, to name a few. Note that these applications are typically difficult via the conventional DSP approaches.

Fig. 4 shows a simplified feedforward artificial neural network. The connection between nodes is mathematically modeled by a linear combination, i.e. weight **a** and bias **b** matrix vectors. A positive weight indicates an excitatory connection, while a negative weight suggests an inhibitory connection. The linear functions make the neural networks easy to be analyzed yet powerful in the computation. In additional to the linear components, neural networks also include various nonlinear operations, typically called activation functions. It regulates the acceptable range of output and also provides the nonlinear fitting capability. Some commonly used activation functions are ReLu, Sigmoid, etc.

In the lecture, we covered the single perceptron, basic neural networks, as well as their building components, such as neurons, layers, activation function, etc. We will also introduce some special layers that are commonly used in the neural networks, such as fully connected layer, convolutional layer, pooling layer, etc. We will also show the back propagation algorithm that is widely used to train the neural network. Students will not only learn how to train a network, but also understand the similarities and differences between the DSP approach and the ML approach.

III. Course Implementation at CSULB

This course is implemented as a graduate level course in the fall semester. It's also open to the advanced undergraduate students. The course materials are delivered by lectures. Each topic takes about 3 weeks for lectures. Quiz is given frequently to reinforce the key concepts. Each topic is followed by a project assignment, which consists of several comprehensive questions. Students can form a group up to 3 to work on the project together. We encourage the collaboration in the project. Near the end of semester, each student is required to give an oral presentation. They can choose either the project they have worked on or their own research topic in the oral presentation. The oral presentation is also evaluated based on the standard of a formal research conference. There are one midterm and one final exam in this course.

A. *DSP topic 1: Sigma Delta Modulation (SDM)*

In the lecture part, we discussed Shannon/Nyquist sampling theorem and the quantization noise in the first week. Then we discussed the effects of oversampling and loop filtering on the quantization noise in the second week. The theoretical analysis was also given. In the third week, we studied the architecture of SDM, and its signal transfer function (STF) and noise transfer function (NTF). We also compared SDM to other analog to digital conversion architectures, such as successive approximation register (SAR) ADC, pipeline ADC in various system specifications, and hardware cost. We concluded that SDM was suitable for applications that required a very high SQNR while a low or moderate sampling rate.

One quiz is given to test the understanding of quantization noise in the analog to digital conversion process. Students need to master the SQNR equation shown in Eq. (1). Moreover, we have developed a SDM design project for this topic. The project is based on the audio signal processing applications. Given the system specs in Table II, student is asked to design a proper SDM architecture and calculate the parameters of each module in Fig. 2. The project manual includes some references that students may use. It guides students with many open questions, for example,

- What is the oversampling ratio in this system according to Table II?

TABLE II
Sigma-delta modulator project specifications

Specifications	Values
Signal-to-quantization noise ratio	100 dB
Signal frequency	146.1 Hz
Sampling rate	99991 Hz

- How many quantization bits are required to support a 100 dB signal-to-quantization noise ratio (SQNR)?
- What the order of loop filter is?
- What is the STF and the NTF in the proposed SDM?

Note that, the above questions may have more than one answer to achieve the specification requirements. For example, the higher the loop filter order is, the more SQNR can be achieved. Similarly, the larger the number of quantization bits is, the more SQNR can be achieved. Students can choose any one that can achieve the specs. We encourage the students to choose their own design. And we also ask students to analyze any potential benefits and risks in terms of system performance and hardware implementation based on their design.

This project can be implemented in either MATLAB source code or Simulink. MATLAB source code generally runs faster, and is usually compatible across different MATLAB versions. Simulink is more intuitive to illustrate the system architecture. Instructors can choose either way based on the actual needs.

B. *DSP topic 2: Adaptive Filtering*

In the lecture part, we discussed the basics of finite impulse response (FIR) filters and infinite impulse response (IIR) filters in the first week. We studied the cost function and gradient descent solver in the second week. The LMS algorithm is analyzed in the third week with emphasis on the closed-loop transfer function.

One quiz is given to test the understanding of transfer function in a filtering system. Students need to master the basics of FIR LPF analysis. Furthermore, we have developed an electrocardiography (ECG) signal processing project for this topic. ECG is one of the most common vital signals in our body. It measures the bio-potential generated by electrical signals that control the expansion and contraction of heart chambers [26]. ECG test can check the heart's rhythm and electrical activity. Though an ideal ECG signal looks simple, the real and raw ECG signal may be corrupted by noise and body motion. For example, 50/60 Hz AC distortion can be seen in normal capture. And body movement can cause a large DC variation over the time. Therefore, it's critical to suppress noise and extract the signals from interference. In this topic, we demonstrate how to obtain vital information from the raw ECG signals.

There are online ECG signal database for free access [28]. It includes record of normal rhythm as well as abnormal ones. Instructors can also use ECG sensors (or PPG sensors) to obtain heart signal real time.

This project is designed in MATLAB. The project manual guides the student to build up an adaptive filtering block step by step rather than using a capsulated function from MATLAB toolbox. Some example questions/tasks are shown below.

- Show the time domain waveform and power spectrum of provided dataset.

- Identify interference, such as power line frequency, slow motion, etc.
- Design a digital filter to mitigate the power line distortion. Show the type of design filter, order, together with the magnitude and phase response.
- Propose the adaptive filter and determine the adaptive rate based on the ECG application.
- What is heart rate based on the R peak detection?
- Analyze the heart rate estimation, and decide which dataset may come from a cardiac dysfunctions patient

C. ML topic 1: Linear Regression

In the lecture part, we discussed various cost function in the first week, such as norm-1 versus norm-2 regulation terms. In the second week, we studied the necessary and sufficient conditions for local optimum in single-variable optimization problems. We further analyzed the least squares problem and the maximum likelihood and least squares solutions in the third week.

One quiz is given to test the understanding of least squares solution to a linear problem. Students need to master the basics of least squares estimator shown in Eq. (4). Moreover, we have developed a curve fitting project in Tensorflow for this topic. The project is to find the response of a black box based on the output observations on the known input. It is implemented in neural networks. The project manual guides the students, step by step, how to use the fully connected layer from Tensorflow [29], and the build-in gradient descent optimizer to form a curve fitting framework. For example, the project first asks to fit a quadratic function with two unknown parameters based on the input and output data. This can be implemented in Tensorflow in a concise way. It's important to explain each of building blocks in the networks, such as input output interface, number of operation nodes, tensors. Note that Tensorflow has a great tool to visualize the neural networks via Tensorboard. It not only shows the flattened network, but also shows the evolution of selected parameters over the time. It's a good opportunity to present the state-of-the-art tools in ML area to the students. Some example questions/tasks are shown below.

- Load the experimental data in the provided Tensorflow template.
- Complete add_layer() function. With given Weights, biases, and **Wx**_plus_**b** (bold means vector or matrix), generate outputs tensor with activation function call.
- Create a neural network that consists of two defined layers. Try a size of 10 for First_Layer output, and ReLu for activation function "tf.nn.relu".
- Implement the cost function using mean square error.
- Test the designed neural network with the provided dataset. Show the cost reduction, and parameters estimation over iterations in Tensorboard.
- Analyze the convergent results. Propose two different approaches that can speed up the convergence. And also analyze potential problems, if any.

The project also includes a session to estimate the response of a higher order nonlinear system. Since the number of system order is unknown, it's an open question to the students how many computation nodes, tensors as well as the output ports to be used. Project manual includes some questions about the trade-off between fitting error reduction and overfitting prevention.

D. ML topic 2: Neural Networks

In the lecture part, we discussed the principals of single perceptron in the first week, including its definition and learning rules. Then we demonstrated the single perceptron was capable of implementing basic logic functions such as logic AND gate, logic OR gate, etc. We also showed its limitations to the linear classification. In the second and third week, we studied the convolutional neural networks. We analyzed the functionality of its building blocks from the signal processing perspective.

One quiz is given to test the understanding of single perceptron in modeling logic XOR gate. Furthermore, we have developed a hand-written digits recognition project for this topic. The project is implemented in the Tensorflow framework. The project manual guides the students step by step to compose a multi-layer fully connected neural network. For digits recognition, there are ten possible outcomes. It's a relative simple yet practical problem that the students can learn to decide the parameters of input and output layers. Students will also learn how to train a neural network based on the MNIST database [30]. Note that, Tensorflow has provided rich APIs to use so that a neural network can be implemented in a few lines in Python. Though the details of the math parts, like how back propagation, gradient descent solver are implemented in the programming are omitted, we open the door to those who are interested in exploring more on a particular subject. We believe this is beneficial to motivate students for future study.

The project manual will guide the students to evaluate the proposed neural networks in terms of prediction precision and computation cost. The students will revisit overfitting problem that has been seen in the previous ML topic.

In addition to the standard MNIST database, we also ask the students to prepare their own hand-written digits. We notice at least three benefits of doing this: (1) students will be truly impressed by their own work; (2) extra signal processing is mandatory to fit the self-prepared images into standard format; (3) students will have a clearer picture on training, validation, and verification phase in the ML area.

Moreover, several graduate and undergraduate students in the class participated a shark research project at CSULB. In this project, a supervised ML method, K-nearest neighbors (KNN), in conjunction with FFT is applied to classify behaviors of sharks based on the data collected by tri-axial acceleration data loggers (ADLs) as shown in Fig. 5 [20-23]. KNN is a simple machine learning algorithm in which, a new data point is classified based on the majority of its k nearest neighbors, k being a user defined hyper-parameter. The algorithm does not use any model to fit the new data sample but rather computes the distance of the data point form its neighbors [19]. Sharks exhibit many behaviors naturally occur in the water, including Resting, Swimming, Feeding, and Non-directed Motion (NDM). Resting is defined as the shark lying motionless on the seafloor. Swimming is featured by the repetitive tail beats and forward movement of the shark. Feeding refers to prey capture, handling, chewing, and head shaking. Movements along the seafloor that could not be identified as Resting or Swimming are classified as NDM. The data collected by the ADL is labeled manually by watching the GoPro video and identifying the shark behavior with the corresponding time-stamps. Every 25 data samples (one second of data) are aggregated as an instance to represent the shark behavior per second for classification.

We use digital signal processing techniques such as FFT and Discrete Cosine Transform (DCT) to represent the accelerometer signal in the frequency domain which reduces the data size required for training the classifier, the computation time and the memory resources needed, in addition to improving model accuracy. The shark behavior is classified into four classes namely Resting, Swimming, Feeding and Non-Directed Motion via KNN algorithms. We compare different combinations of time and frequency domain data on the performance of the algorithm. It is shown that the transform domain data considerably improved the accuracy of the classifier.

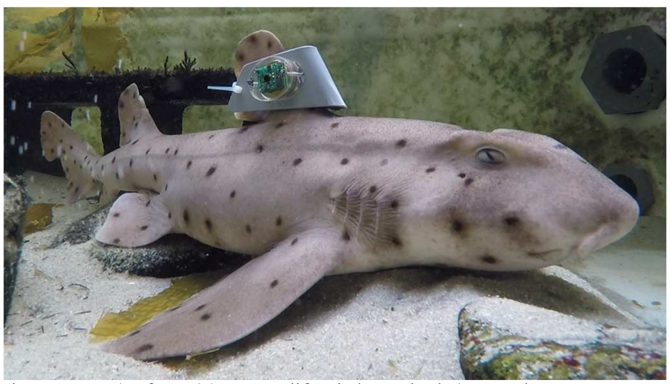

Fig. 5. ADL (Cefas G6a) on a California horn shark (Heterodontus francisci).

IV. Survey Design and Results

The assessment of the modules and projects is based on a survey with students who have taken this course at CSULB. There are a total of 56 people completed the survey. Table III list the background of students taking this course. The majority of students are undergraduate, counting up to 91.1%. All undergraduate students are either in the third or the fourth year of Bachelor of Science program. The rest 8.9% are graduate students.

We further surveyed how much background knowledge of DSP and ML did students have before taking this course. Table IV summarizes the responses. Though all students indicated they meet the minimum prerequisites of this course, we can see the amount of background knowledge on DSP and ML is quite different. There is 85.7% of feedback indicated sufficient amount of DSP background, while there is only 19.6% indicated sufficient ML background. In contrast, nobody claimed little knowledge on DSP, but there was 40.2% claimed so on the ML domain knowledge. This significant difference is likely due to the fact that this course is implemented in the Electrical Engineering curriculum. Students taking this course are more likely to have better background knowledge on DSP than ML due to past courses. We also noticed this difference in

TABLE III: Background of students in the survey.

Questions	% of Responses (n=56)
Undergraduate or Graduate	91.1% Undergraduate 8.9% Graduate
Do you have background on DSP	100%
Do you have background on ML	39.3%

TABLE IV: Survey question: how much background of DSP and ML do you have before taking this course?

Domain knowledge	% of Responses (n=56)
DSP	85.7% Sufficient 14.3% Moderate 0% Little
ML	19.6% Sufficient 40.2% Moderate 40.2% Little

TABLE V: Survey question: which module in this course contributes most to your learning success?

Module	% of Responses (n=56)
Lecture	37.5%
Project	48.2%
Homework	14.3%

the teaching practice. Hence, we spent more time in exploring the similarities and differences between DSP and ML, and put more resources on the ML projects, such as tutorials on Python programming, Tensorflow, etc.

This course includes both lectures and assignments. We investigated which part was considered most helpful to the students. Table V summarizes the students' feedback on survey question "which module in this course contributes most to your learning success?" There are 37.5% and 48.2% of students chose lecture and project, respectively. Many students also commented that the lectures and project assignments expanded their vision on the generalities of the technology, the interesting applications, and the latest advancements, especially the ML part. There are 14.3% of students chose homework. In the comments, they highlighted homework was most helpful in preparing exams.

We further surveyed if this course has benefited the students from the professional development. Table VI summarizes the students' feedback on survey question "will this course help you professionally after graduation, particularly compared to other courses you have taken?" All graduate students indicated a positive feedback. Especially, they commented that this course helped them prepared a clear roadmap on the future research. Some students also commented that this course introduced very practical tools that were popular in the industry, such as Python and Tensorflow, which would surely improve their competence. However, the feedback from undergraduate students was mixed. Only 64.7% gave a positive response, while 25.5% gave negative response, and 9.8% said not sure. In

TABLE VI: Survey question: will this course help you professionally after graduation, particularly compared to other courses you have taken?

Professional development	**% of Responses (n=56)**
Yes	64.7% Undergraduate 100% Graduate
No	25.5% Undergraduate 0% Graduate
Not sure	9.8% Undergraduate 0% Graduate

TABLE VII: Survey question: has this course increased your desire to pursue academic research on DSP and ML in the future?

Motivation to academic research	**% of Responses (n=56)**
Yes	49.0% Undergraduate 100% Graduate
No	35.3% Undergraduate 0% Graduate
Not sure	15.7% Undergraduate 0% Graduate

the positive feedback, students had highlighted the practice of Python and ML applications. In the negative feedback, they had mentioned Java as the most helpful course and tools for job hunting. Some also commented the missing project solution had compromised their learning achievement compared to other courses.

Since this is an introduction course to advanced DSP and ML, we also surveyed if this course had motivated the students to continue the study or research on the DSP and ML in the future. Table VII summarizes the students' feedback on survey question "has this course increased your desire to pursue academic research on DSP and ML in the future?" All graduate students indicated a positive feedback to this. Especially, they commented that this course was helpful to their research thesis on power system, communication, etc. However, the feedback from undergraduate students was mixed. Only about 49.0% gave a positive feedback. Among the negative feedback, they have mentioned there was no further academic or research plan. Some mentioned other professional area, such as database management, web design, etc. Among the "not sure" feedback, they mentioned no adequate knowledge to make a plan yet for future.

V. CONCLUSION

We presented the design and implementation of a course that integrated both advanced DSP to ML. It showed that this course can benefit from the strategic introduction of desirable technology in the curriculum efficiently. The design of the project experience and assessment combined with the use of MATLAB and Python in a single course allowed us to deploy and verify the usefulness of a DSP/ML integrated course with several projects in teaching EE and CS students. Moreover, the high level of student involvement in, and enthusiasm for, course participation provides uncommon enrichment of the students' education and professional development and significantly enhances their career opportunity. These hands-on projects ensure that an effective pathway exists for students from diverse backgrounds to receive the education necessary to begin a career in DSP/ML.

REFERENCES

[1] S. Wood and S. Kemnitzer, "First year DSP education in the context of ece curriculum reform," in *Proc. of 2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, Marco Island, FL, pp. 425-429.

[2] H. Purwins, et al., "Deep Learning for Audio Signal Processing," *IEEE J. Sel. Topics. Signal Process.*, vol. 13, no. 2, pp. 206–219, May 2019.

[3] M. Kandlhofer, et al. "Artificial intelligence and computer science in education: From kindergarten to university," in *Proc. IEEE Frontiers Educ. Conf.*, Eire, PA, Oct. 2016, pp. 1-9.

[4] X. Hu, et al., "A Parallel JTAG-based Debugging and Selection Scheme for Multi-core Digital Signal Processors," in *Proc. IEEE Int. Conf. Safety Produce Informatization*, Chongqing, China, Dec 2018, pp.527 -530.

[5] Y. Weng, "Recent Advances in DSP Techniques for Mode Division Multiplexing Optical Networks with MIMO Equalization: A Review," *Appl. Sci.* **2019**, *9*(6), 1178; https://doi.org/10.3390/app9061178.

[6] N. D. Sidiropoulos, et al., "Tensor Decomposition for Signal Processing and Machine Learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, July 2017.

[7] A. D. Sarwate and K. Chaudhuri "Signal Processing and Machine Learning with Differential Privacy: Algorithms and challenges for continuous data," *IEEE Signal Process. Mag.,* pp. 86-94, Sept. 2013.

[8] S. Kolouri, et al., "Optimal Mass Transport: Signal processing and machine learning applications," *IEEE Signal Process. Mag*., pp. 43-59, July 2017.

[9] "Bob: A Free Signal Processing and Machine Learning Toolbox for Researchers," in *Proc. 20th ACM Int. Conf. Multimedia*, Nara Japan, Oct 2012, *pp. 1-4*.

[10] G. S. Randhawa1, et al., "ML-DSP: Machine Learning with Digital Signal Processing for ultrafast, accurate, and scalable genome classification at all taxonomic levels," *BMC Genomics,* **20**, Article number: 267, pp. 1-21, April 2019.

[11] U. S. Shanthamallu, et al., "Introduction Machine Learning in Undergraduate DSP Classes," in *Proc. IEEE ICASSP 2019,* Brighton, UK, May 2019, pp. 7655-7659.

[12] H.-G. Yeh, et al, "Wavelet and Denoising Techniques for Real-Time HIF Detection in 12-kV Distribution Circuits," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4365-4373, Dec. 2019.

[13] M. Tadayon and G. J. Pottie, "Predicting Student Performance in an Educational Game Using a Hidden Markov Model," *IEEE Trans. Education,* vol. 63, no. 4, pp. 299–304, Nov. 2020.

[14] A. J. Parmar, "Bridging gaps in engineering education: Design thinking a critical factor for project based learning," in *Proc. IEEE Frontiers Educ. Conf.*, Madrid, Spain, 2014, pp. 1–8.

[15] S. R. Brunhaver, et al., "Bridging the gaps between engineering education and practice," *U.S. Engineering in a Global Economy*, 1st ed. Cambridge, MA, USA: Nat. Bur. Econ. Res., 2018, pp. 129–163.

[16] S. L. Benton, D. Li, and W. H. Pallett, "In higher education, class size matters," presented at the Annu. Meet. Amer. Psychol. Assoc. Conf., Honolulu, HI, USA, 2013, pp. 1–32.

[17] M. A. Richards, *Fundamentals of Radar Signa Processing*, McGraw Hill, 2014,

[18] D. C. D. Chang, H.-G. Yeh, and Qubao He, "Coherent Power Combining on Spacecraft via Wavefront Multiplexing Techniques," in *Proc. the IEEE Int. Conf. on Computing, Networking and Commun. (ICNC) Workshop*, San Diego, CA, pp.268-272, Jan. 2013.

[19] S. J. Preece, et al., "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomedical Engineering*, vol. 56, pp. 871-879, 2009.

[20] S. Karan, et al., "Classification of Shark Behaviors using K-Nearest Neighbors," in *Proc. 2019 IEEE Green Energy and Smart Systems Conference*, Long Beach, CA, 2019, pp. 1-6.

[21] Y. Yang, et al., "Feature Extraction, Selection and K-Nearest Neighbors Algorithm for Shark Behavior Classification Using Imbalanced Dataset," Early Access, *IEEE Sensors Journal*, pp. 1-10, Nov. 2020.

[22] W. Zhang, et al., "Deep Convolutional Neural Networks for Shark Behavior Analysis," in *Proc. IEEE Green Energy and Smart Systems Conference*, Long Beach, CA, 2019, pp. 1-6.

[23] I. M. Ali, C. J. Lee, H.-G. Yeh, S. Karan, Y. Yang, W. Zhang, E. Meese, C. G. Lowe "Improvement of Classification of Shark Behaviors using K-Nearest Neighbors," in *Proc. IEEE Green Energy and Smart Systems Conference*, Virtual, Nov 2020, pp. 1-7.
[24] J. Culley, et al., "System Design for a Driverless Autonomous Racing Vehicle," in Proc. 12th Int. Symposium on Commun. Systems, Networks and Digital Signal Process., July 2020, Porto, Portugal. July 2020, pp. 1-6.
[25] A. P. James, et al., "HTM Spatial Pooler With Memristor CrossbarCircuits for Sparse Biometric Recognition," *IEEE Trans. Biomedical Circuits and Systems*, vol. 11, no. 3, pp. 640- 651, June 2017.
[26] K. S. Bhavsar, H.-G. Yeh, P. Ayala, "An Aided Information to Characterize ECG Signals as Normal or Abnormal," in *Proc. Int. Symp. Circuits and Systems (ISCAS)*, Baltimore, MD, May 2017, pp. 682-685.
[27] Anaconda data science toolkit. Installation reference webpage: https://www.anaconda.com/products/individual
[28] Electrocardiogram Vigilance with Electronic data Warehouse II (ECG-ViEW II), freely available at http://www.ecgview.org.
[29] TensorFlow: A System for Large-Scale Machine Learning. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
[30] Y. LeCun and C. Cortes, MNIST handwritten digit database, webpage: http://yann.lecun.com/exdb/mnist/