Module 4

Working with Neural Networks

- What is a neural network?

  a system of networked biological, artificial, or simulated neurons that perform some function usually attributed to intelligence, such as tasks involving memory, information processing, or pattern recognition

- What is a neuron?

  a node in a connectionist information processing system

- To what can the beginning of modern neural network theory can be attributed?

  the McCulloch–Pitts Theory of Formal Neural Networks (1943)

- What tasks are ANNs well suited for?

  tasks that involve high levels of complexity and uncertainty, like prediction and classification

- Are ANNs more accurate in their predictions than standard statistical prediction techniques?

  Yes

- What are the 3 parts of an ANN?

  the input layer, one or more hidden layers, and the output layer

- What is the depth of an ANN?

  a measure of how many hidden layers it has

▼ What are input values multiplied by?

weights

▼ How are weights updated?

Weights are usually initially randomized, and a process called **gradient descent** is used to update these weights after the gradient of the model's **error rate/loss function** is calculated during a process called **backpropagation.**

▼ What is gradient descent?

an optimization algorithm used to find the values of parameters (coefficients) of a function that minimizes a cost function (cost)

▼ What is a cost function?

- Cost functions are used to estimate how badly models are performing.
- A cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y.
- This is typically expressed as a difference or distance between the predicted value and the actual value.
- The cost function (referred to as *loss* or *error*) can be estimated by iteratively running the model to compare estimated predictions against “ground truth” — the known values of *y*.

▼ What is the objective of a machine learning model?

find parameters, weights or a structure that **minimize the cost function**

▼ What is backpropagation?

- It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration).
- Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.
- Backpropagation in neural network is a short form for “backward propagation of errors.”

▼ How does backpropagation work?

1. Inputs X, arrive through the preconnected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs
5. **Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.**
6. Keep repeating the process until the desired output is achieved.

▼ Why is a constant bias value added?

to adjust the input to the data and shift the output of the activation function towards the positive or negative side

▼ What is a bias value in an ANN?

- Bias is **like the intercept added in a linear equation**.
- It is an additional parameter in the neural network which is used to adjust the output along with the weighted sum of the inputs to the neuron.
- Thus, bias is a constant which **helps the model in a way that it can fit best for the given data.**

▼ What is the purpose of an activation function?

- It decides whether a neuron should be activated or not
- This means that it will **decide whether the neuron's input to the network is important or not in the process of prediction** using simpler mathematical operations

▼ What are feedforward neural networks?

- Feedforward neural networks are one of the simplest types of neural networks because they are **unidirectional**.

- This means they do not use backpropagation, do not use cycles or loops of any kind, and **maintain static weights**.
- A **perceptron** is an example of a feedforward network; a single layer of perceptrons is functionally identical to a logistic regression model even when using a sigmoid activation function.
- Unmodified feedforward networks are less common in modern applications, and are **not considered to be deep.**

▼ What are convolutional neural networks or CNNs?

- Convolutional neural networks or CNNs are composed of **convolutional** layers, meaning **two functions are combined to create a third**, in this case, a **feature map**.
- CNNs are efficient, easy to train, and have wide application in **video**, **image**, and speech applications.

▼ What are recurrent neural networks or RNNs?

- Recurrent neural networks, or RNN, **implements memory functions** (such as in long short term memory, or **LSTM**), always **looping some of its outputs back to its hidden layers**.
- RNNs are crucial for **sequenced data and time-series data**, and are commonly used in **natural language processing**, speech recognition and translation, and text and sentiment classification.

▼ What are autoencoders and transformers, and Sequence to Sequence?

- **Autoencoders** (a type of **encoder-decoder**) are a type of unsupervised model that has an output layer with the same number of units as the input layer, often with a hidden layer with less units.
- Encoder-decoders are a special type of **LSTM RNN**. When two of these are combined, you have a **Sequence to Sequence**, or **Seq2Seq**, model.
- When attention mechanisms are used in place of RNN, a much more robust model called a transformer is created.

- Both RNN-based Seq2Seq models and transformers are widely used in neural machine translation, NLP, and chatbot technology.

▼ What is a Generative Adversarial Network (GAN)?

- a set of two networks, one *generative* and one *adversarial*, combined
- the generative network produces iterations of a certain output (often an image but can be of many types)
- the adversarial network tries to use classification methods to determine whether the image is real or fake
- as soon as the generative network is able to fool the adversarial network, the GAN produces its output
- this is used today in many applications, notably to generate realistic images of humans or other objects that are difficult to impossible for humans to distinguish as not being "real"

▼ What are Radial Basis Function (RBF) networks?

- they work on principles similar to K-NN regression models
- they are extremely efficient as universal approximators
- they can utilize alternative activation functions such as Gaussian, multi-quadratics and inverse multi-quadratics, or a proposed function called Square-law based RBF kernel (SQ-RBF) which eliminates the exponential term as found in Gaussian RBF

▼ Neurons can not fire at a rate...

less than zero

▼ Neurons can not fire faster than what?

- a certain rate
- this problem is solved using a nonlinearity between one and zero (fire and not-fire)
    - the precise nature of this nonlinearity depends on the activation function used, but can be understood as a range of probabilities

▼ What are sigmoid activation functions?

they are a popular and commonly used function that normalizes the output of each neuron to a floating point number between 1 and 0

▼ What problem does the sigmoid function have?

the vanishing gradient problem - meaning there is a plateau beyond which learning becomes difficult

▼ What is a hyperbolic tangent activation function or TanH?

- When you take a sigmoid activation function and center it to zero, it becomes a hyperbolic tangent activation function, or TanH.
- A form of TanH called **penalized tanh** has shown high stability and accuracy that, along with Swish, bested other functions

▼ What is the Swish activation function?

- Swish is a new activation function discovered by Google.
- It is more accurate and can be used in deeper models than ReLU but offers a similar level of computational efficiency.
- Swish uses value ranges of infinity in both directions.
- Swish outperforms almost all other activation functions in prediction accuracy except penalized tanh.

▼ What are ReLU and Leaky ReLU?

- Rectified Linear Unit (ReLU) is a very efficient nonlinear activation function that is linear in the positive axis (yes, it is still nonlinear).
- This leads, however, to the dying ReLU problem, where nodes with weights and biases that are not updated eventually never get activated and become "dead".
- Leaky ReLU attempts to solve this problem by transforming negative values into a small linear component of the x axis.

▼ What happens if you choose not to use an activation function?

- That would give you an output determined solely by the weights and biases, and would be a linear function, essentially giving you a fancy linear regression model, as the network layers would collapse into one anyway.
- This precludes linear activation functions from being used for more complex tasks or learning from experience, and is why you need a nonlinear activation function for an ANN.

▼ Is a deeper network better?

- This is often the case. It is debatable whether depth is more important than width, and the conclusion is often that it heavily depends on the use case.
- But there do exist simple tasks where wide networks excel over deep ones, particularly in cases where inputs are predictable and repetitive, resulting in stability.
- Networks without depth have a hard time generalizing, however.

▼ What are tensors?

- a type of generalized vector (examples include plane vectors, covectors and linear operators) represented as n-dimensional arrays of base data types

▼ What is the main object you manipulate in TensorFlow?

the tf.Tensor object representing a tensor computation