**Machine Learning:**

Slide 2:

So what is machine learning?

Slide 3:

Machine learning is a branch in AI that uses data and algorithms to make predictions. In other words, it’s just maths.

Slide 4:

There are 4 types of machine learning. The 1st type is called supervised learning and it uses labelled data. The 2nd type is called semi-supervised learning and it uses both labelled and unlabelled data. The 3rd type is called unsupervised learning and it uses unlabelled data. The 4th type is called reinforcement learning and it uses feedback. It’s a bit like training a dog, you reward it if it’s doing something good. We’re actually doing a workshop on reinforcement learning at the end of this semester, so please do come if you’re free, we have a lot in store for you guys.

Slide 5:

Today we’re going to be focusing on supervised learning. Here’s an example of labelled data. Each row in this dataset is called an observation. In this case, an observation is a passenger on the Titanic. Each observation has a set of features. In this case, each observation has a name, passenger class, sex, age and so on. Each observation also has a target variable. This is the value that we’re trying to predict. In this case, it’s whether a passenger survived the Titanic or not.

Slide 6:

So what is a machine learning model? A machine learning model is the algorithm that makes the predictions. It takes the features from an observation and tries to predict the observation’s target variable. If we have a good model, then it should predict the target variable correctly.

Slide 7:

So what can a machine learning model do? A machine learning model can be used to predict a quantity. This is called regression. Or it can be used to predict a class. This is called classification. Any questions so far?

**Neural Networks:**

Slide 8:

Okay, so the model that we’re going to be building today is called a neural network.

Slide 9:

A neural network is made up of layers of neurons, just like our brain. Each one of these circles here is a neuron. A neural network has an input layer. The neurons in this layer only contain the observation’s features. Each neuron in this layer passes its feature to each neuron in the next layer. A neural network can then have hidden layers. If it does, it’s called a deep neural network. Each neuron in these layers processes the data that it receives and passes it on to each neuron in the next layer. A neural network then has an output layer. Each neuron in this layer provides a final prediction using the data that it receives.

Slide 10:

Here’s a real life example. Here, we’re using a neural network to classify a plant based on it’s petal length, petal width, sepal length and sepal width. There are 3 final outputs. Each is a different type of plant. The prediction 0 means that it’s not that plant and the prediction 1 means that it is that plant.

Slide 11:

Okay, so how does a neuron work? As we just said, a neuron receives the output of each neuron in the previous layer as its inputs. Each of these inputs is weighted by some value. These weights determine the strength of the inputs. The neuron multiplies each input by its weight and adds them all together. The neuron then adds a bias to this. A bias determines the strength of the neuron. Each neuron has its own bias. The neuron then feeds this into an activation function. I’ll explain what an activation function is in a bit. The value returned by the activation function is the output of the neuron.

Slide 12:

If we take a step back, we can see that each neuron in the hidden layers and the output layer performs these operations. The sum symbol represents the sum of the weighted inputs and bias and the function symbol represents the activation function. Any questions?

**Activation Functions:**

Slide 13:

Okay, so why do we use an activation function. Without an activation function, our neural network becomes linear, as it would only perform multiplications and additions. However, most real-world data is non-linear. For example, the relationship between house price and house size or the relationship between income and purchases. If neural networks had no activation function, then they would fail to learn the complex non-linear patterns in real-world data. We’re going to look at 3 activation functions.

Slide 14:

This is the sigmoid activation function. It’s useful for binary classification as it maps a value to a value between 0 and 1. Binary classification is when we’re trying to classify something using only 2 classes. For example, if an email is spam or not spam or if an animal is a cat or a dog. So 0 could represent one class and 1 could represent another class.

Slide 15:

This is the tanh activation function. It maps a value to a value between -1 and 1.

Slide 16:

This is the RELU activation function. It maps negative values to 0 and positive values to themselves.

Slide 17:

Okay so we’ve covered what machine learning is and the model that we’re going to be building later. We’re now going to take a quick break and have some pizza.

**[Give out pizza]**.

**Learning:**

Slide 18:

We’re now going to go over how our model learns, before we build it.

Slide 19:

So how do we know if we have a good model or not? We use loss. Loss is a measure of a machine learning model’s performance on a set of data. The lower the loss, the more our model’s predictions match reality. We’re using the mean square error as our loss function here, as it’s very easy to understand and it’s commonly used. It’s used for regression tasks. MSE takes a prediction that our model makes and the value that it’s trying to predict, it then subtracts one from the other and squares the result. It does this for each observation in that set of data and then calculates the average. This is actually the MSE for a single output neural network, the MSE for a multi output neural network is slightly different to this.

Slide 20:

Okay, so what do we do if we have a bad model? We train it. Training adjusts our model’s parameters, in order, to minimize its loss on a set of data. This set of data is called the training set. In the case of neural networks, our model’s parameters are the weights and biases.

Slide 21:

Okay, so how do we know if training has improved our model? We test it. Testing uses our ML model’s loss, on an unseen set of data, as an estimate of its general performance. This unseen set of data is called the testing set. Any questions?

**Gradient Descent:**

Slide 22:

**[Skip Slide]**.

Slide 23:

An optimizer is an algorithm that trains a machine learning model. Most optimizers are some variation of gradient descent. However, not all are.

Slide 24:

So what is gradient descent? Gradient descent finds the minimum value of a function f(x) by iteratively adjusting its input x. Here is the equation for gradient descent. It calculates a new value of x by subtracting the derivate of f(x) from the old value of x. It actually also multiplies the derivative of f(x) by a learning rate, this controls how much we adjust x by. The exponent here represents the number of the iteration and not an actual exponent. So xt+1 is the result of the t+1th iteration. We apply gradient descent until we converge on an x value. The x value that we converge on gives us the minimum value.

Slide 25:

The graph here represents gradient descent. Where each jump represents an iteration. As you can see, it’s better to use a smaller learning rate, as you tend to overshoot the x value if you use a big learning rate.

Slide 26:

We can actually re-write our loss function to take each weight and bias, in our neural network, as an input. We can do this, as our loss function uses each prediction, that our neural network makes, to calculate the loss and our neural network comes to these predictions using its weights and biases. So, technically, our loss function uses each weight and bias, in our neural network, to calculate the loss.

Slide 27:

Rewriting our loss function like this allows us to use gradient descent to minimize our neural network’s loss, on the training set, by iteratively adjusting each weight and bias. In other words, this allows us to use gradient descent to train our model.

Slide 28:

The graph here is a visual representation of this. Any questions?

**Backpropagation:**

Slide 29:

Okay, this is the last bit of information before we start building our neural network.

Slide 30:

So we can use gradient descent to train our model.

Slide 31:

However, gradient descent requires us to calculate the derivate of our loss function with respect to each weight and each bias in our neural network. So, how do we do this?

Slide 32:

Well, we use backpropagation. Backpropagation is when we go back through our neural network and calculate each of these derivatives for gradient descent. So gradient descent relies on backpropagation.

Slide 33:

Calculating these derivatives is actually quite complex. So we calculate them using the chain rule, as each of these 3 derivates is much easier to calculate. Don’t worry, we’re not going to calculate them. Any questions?

**Recap:**

Slide 34:

So, to recap what we’ve learnt.

Slide 35:

We’ve learnt what machine learning is. We’ve learnt how neural networks work. We’ve learn about the different types of activation functions. We’ve learnt what training and testing a machine learning model means. And we’ve learnt how we can use gradient descent and backpropagation to train a neural network.

**Workshop Info:**

Slide 36:

Now onto the fun part, let’s build a neural network.

Slide 37:

Okay, so we’re going to try and train a neural network to predict whether a passenger on the Titanic survived or not. We’re using the Titanic dataset, that I used earlier as an example, to do this.

Feature Descriptions:

-Pclass is the passenger class of the passenger.

-SibSp is the number of siblings and spouses that the passenger has on board.

-ParCh is the number of parents and children that the passenger has on board.

Datset Link:

<https://www.kaggle.com/competitions/titanic/data?select=test.csv>