Neural Network Maths

Thomas Simm

Table of contents

```
#hide
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
```

Gradient Descent

Gradient descent is an algroithm used to determine the value of parameters that are used to map input variables to target variables. The steps are: - Determine a loss function that relates predictions and actual values, i.e. a function that determines how good the predictions are - N.B. the function needs to be of a form to encourage gradient descent - Find the gradient of the loss function with respect to the parameters - Update the weights based on this gradient

For example, in the figure below - if the weight W=0 then the Loss is 1 and the gradient is -1.0. In this case we'd want to increase the value of the weight W towards it's minimum 0.5 - if the weight W=1 then the Loss is also 1 but the gradient is +1.0. In this case we'd want to reduce the value of the weight W towards it's minimum 0.5 - if the weight W=0.5 then the loss is close 0 and so is the garient, in this case we wouldn't want to change the weights by much

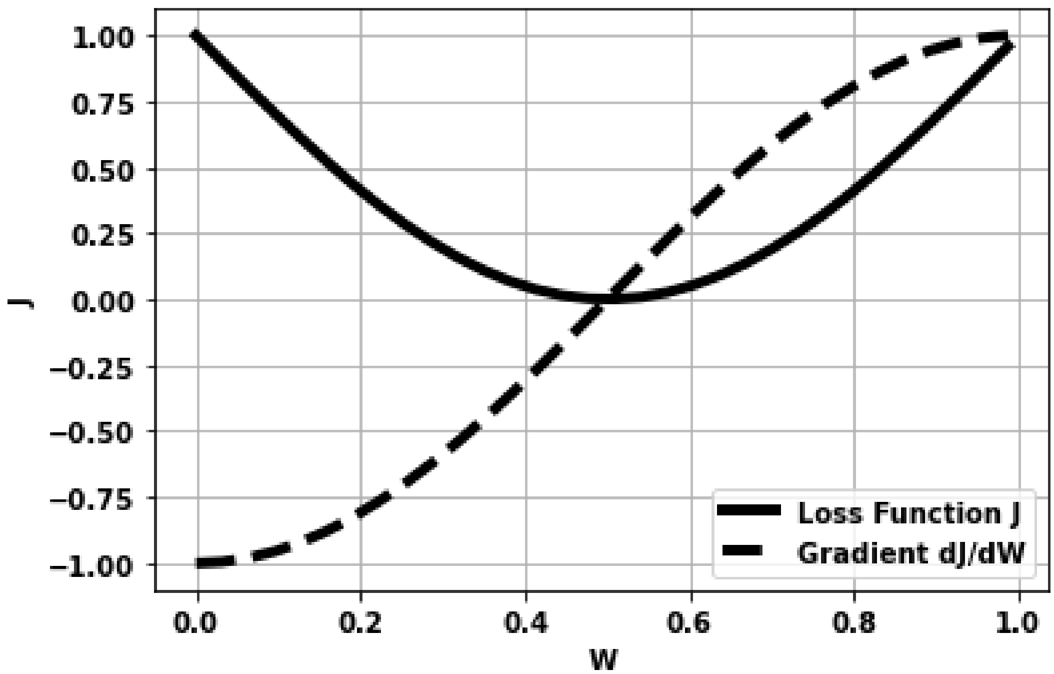
i.e. the weights can be updated by a formula of the form below to get the best fit:

$$W_{i+1} = W_i - const.\frac{dJ}{dW}$$

> Gradient descent is not the only optimisation method to determine parameters but is an easy one to understand.

```
#hide-input
xx=np.arange(0,np.pi,.1)
```

```
plt.plot(xx/np.pi,1+np.sin(-xx),'-k',linewidth=4)
plt.plot(xx/np.pi,-np.cos(-xx),'k--',linewidth=4)
plt.legend(['Loss Function J','Gradient dJ/dW'])
plt.xlabel('W')
plt.ylabel('J')
plt.grid(True)
```



Some definitions.

- `X` are the input variables
- `Y` the target variables
- $\hat{y}$ is the prediction
- The loss function `J` is what we are trying to minimise and is the sum of $Y - \hat{y}^2$. For simplicity we'll remove the summation below.
- `m` and `b` are the parameters we are looking to obtain

$\hat{Y} = mX + b$

$Error = \hat{Y} - Y$

$J(m, b) = Error^2$

$Error = mX + b - Y$

What we want to do is update `m` and `b` so that `J` is reduced and the predictions is better. That is for `m` we obtain a new value (1) from the previous one (0) as follows: