Assignment 2: Neural Networks
Summary

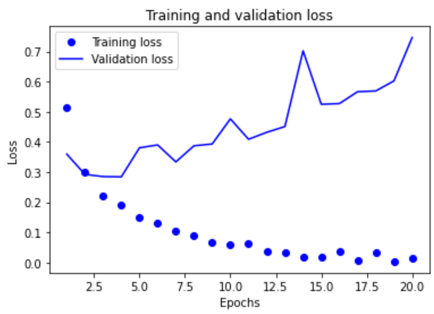
I made the use of two concealed levels. Experiment with one or three concealed levels to see how they influence validation and accuracy rate.

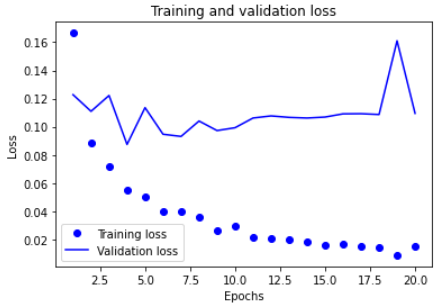
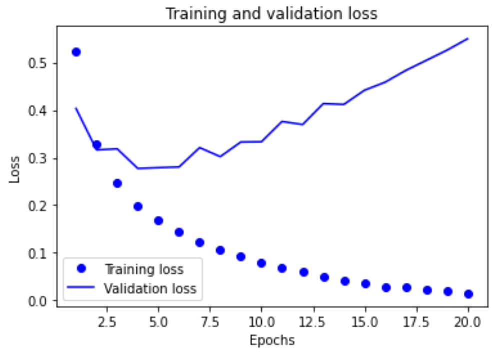
Significance:

- In my earliest approach to the Problem, I learned about the import of the Kera's Sequential model, which is a collection of layers for constructing neural network models.
- It includes the necessary components for designing an neural network, such as layers, Dense, Dropouts, and Regularises.
- I tested a efficacy of two, three, and six-layered neural networks with 16, 64, and 64 concealed neurons.

Affects validations and test accuracy for a given point

I used layers to an One important thing I noticed is that no matter how often layers we added up, this will learn and function nearly identically once it hits the threshold.

Combinations	Training accuracy	Validation accuracy
2 dense layers 16 hidden units Tanh activation function Optimizers = adam	0.84	0.8401 

3 dense layers 64 hidden units Tanh activation function Dropouts (0.5) Regularizers Optimizers = adam	0.87	0.8733 
6 dense layers 64 hidden units Tanh activation function Dropouts (0.5) Optimizers = adam regularizers	.0.8461	0.8767 

model.add(Dense(64, activation="tanh"))

I've included the component unit value64 in the given model description.

• The previous line indicates that a new hidden layer with 64 compact units was added a using the tanh activation function.

• Because once I state 64 hidden layers, we can infer that 64 neurons are f in the stack to acquire the data, which is in the shape of vectors.

• The input vector is also known as the transfer function; if the function's output range is restricted, simply sigmoid squash the number to 1 or higher. They are functions that are not linear.

the training set and validation BCE are used for three level of 16 nodes.

The decline for BCE begins at 0.42 and decreases to 0.42 at the 20th era.

The confirmation loss for BCE begins at 0.64 and increases to 0.87 at the 20th era.

For the unit value 32 the training set and validation

BCE is used in 32 nodes with three levels.

The loss for BCE begins at 0.36 and decreases to 0.87 at the 20th century.

The confirmation loss for BCE peaks at 0.59 and increases to 0.87 at the 20th era.

For the unit value layer 64 the training set and validation

BCE is used in 64 nodes with three levels.

The loss for BCE begins at 0.79 and continues to rise until the 20th century, when it reaches 0.87

The confirmation loss for BCE begins at 0.43 and increases to 0.61 at the 20th era.

For the 3rd step

I using the MSE loss function instead of binary_crossentropy.

The binary Compiling the models

For the tanh the training set and validation the Values are increased

The decline for BCE begins at 0.12 and decreases to 0.1008 at the 20th era.

The loss Precision for BCE begins at 0.71 and increases to 0.99 at the 20th era.

The confirmation loss for BCE begins at 0.12 and decreases to 0.1008 at the 20th era.

The confirmation Accuracy for BCE begins at 0.84 and increases to 0.87 at the 20th era.

model.adding(Dropout(0.5))

• Dropout technique with a rate of 0.5 is used after each concealed layer in this instance.I employing L2 regularization with a 0.001 coefficient on the weights of the concealed levels.

• I attempted using L1 and L2 regularizes, but they have little impact and actually degrade performance. I believe the model is saturated, and the highest validation precision we can achieve is in the 86-87 percentile.

• To test the efficiency measures on the loss, I replaced binary Crossetrophy with mean square errors.

• Because of the diminishing gradient issue, the relu functions is superior to the sigmoid and tanh functions. Tanh accomplishes the same as tanh in this situation.

As we can see, using dropout regularization improves validation accuracy significantly more than the prior models without regularization. This implies that the model was previously overfitting to the training data, and dropout regularization assisted in reducing overfitting and improving generalization capability.