In this assignment I contructed a rather basic network:

“””

model = Sequential()

model.add(Dense(20, activation='relu', input\_dim=26))

model.add(Dense(10, activation='relu'))

model.add(Dense(5, activation='softmax'))

model.add(Dense(1))

“””

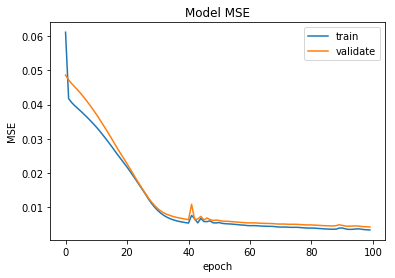
It’s a sequential model with 4 fully connected layers (not including input): a layer taking the input and outputting 20 nodes using the relu activation function, a layer taking those 20 and outputting 10 nodes using relu, a layer taking those 10 and outputting 5 nodes using softmax, and a final linear layer to output a single result.

“sgd = SGD(lr=0.1)”

Stochastic gradient descent with a learning rate of .1 was used.

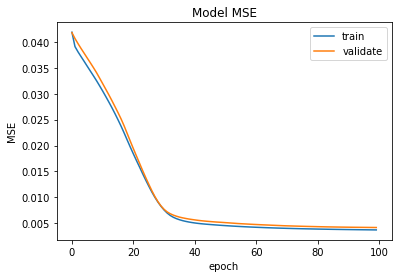
The loss function was mean squared error.

Out of the input data, 800 was partitioned for training data, 200 for validation, and the rest for test data.

When fitting the model, I used 10 validation steps, 44 steps per epoch, and 100 epochs. The result can be seen here:

With a MSE on the test set of 0.004.

When the validation split is adjusted from .2 to .25 we get the following results:



With an MSE on the test set (same test set as before) of 0.003