

Machine Learning Summative

wbbz74

22nd March 2019

1 Image Classifier

The base network consists of a simple, 3 layer fully connected network containing only 1 hidden layer. This performs poorly as there is little opportunity for the network to learn deep and abstract representations by using a single hidden layer. Additionally, spatially related information is not considered together—a vital aspect of an image classification task, where neighbouring pixels need to be considered as a unit in order to identify items in given images. After around 50 epochs, this network only shows accuracy of around 15%, also showing signs of overfitting as the training becomes better classified than the test set.

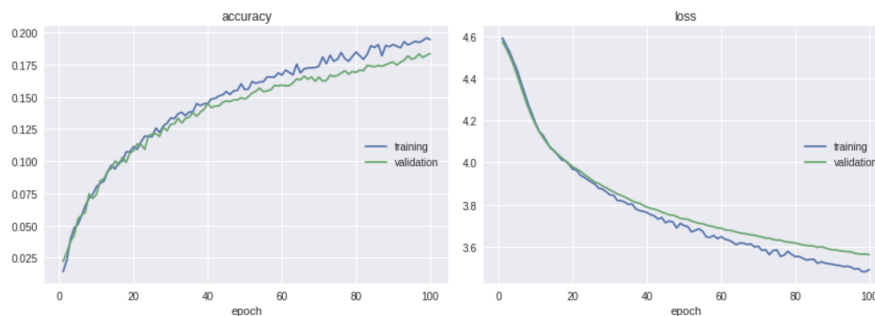


Figure 1: Initial accuracy, 3 fully connected layers (1 hidden) with ReLU normalisation.

To remedy this, we replace the pure linear structure of the network with a convolution based network. Convolutional Neural Networks (CNNs) are able to better learn representations based on a number of spatially related inputs, allowing for better generalisation of image data. This is achieved by learning convolution masks and which reduce the dimensionality of the image.

My first modified network consisted of 2 convolutional layers, each followed by a max pooling layer (2x2). Then, flattening this output, it is passed to two hidden fully-connected layers before the final classification layer. As shown in Figure 2, after 100 epochs the network shows far fewer signs of overfitting as the accuracy climbs past 20%.

```
class BetterNetwork(nn.Module):  
  
    def __init__(self):  
        super(BetterNetwork, self).__init__()  
        self.conv1 = nn.Conv2d(3,6,5)  
        self.pool1 = nn.MaxPool2d(2,2)
```

```

self.conv2 = nn.Conv2d(6,12,3)
self.pool2 = nn.MaxPool2d(2,2)
self.fc1 = nn.Linear(in_features=432, out_features=512)
self.fc2 = nn.Linear(in_features=512, out_features=100)

def forward(self, x):
    x = self.pool1(F.relu(self.conv1(x)))
    x = self.pool2(F.relu(self.conv2(x)))
    x = x.view(x.size(0), -1) # flatten
    x = self.fc1(x)
    x = self.fc2(x)
    return x

```

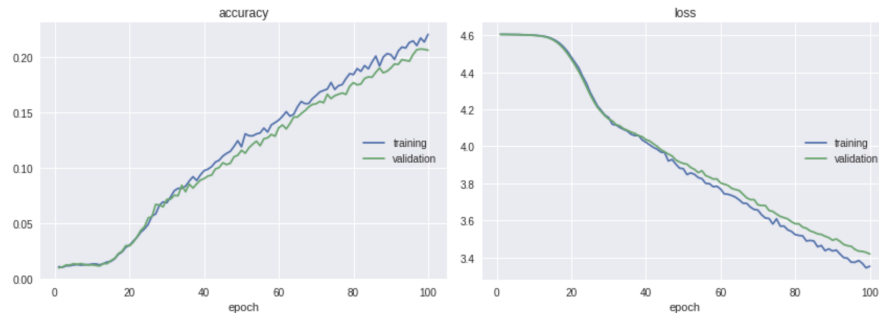


Figure 2: Initial convolution based network (2 conv2d layers, 2 maxpooling, 2 fully-connected layers).

In order to try and obtain better results from the existing network, I increased the batch size from 16 to 32. This should allow for better generalisations to be learned at once, as more training examples will now be considered before weights and biases are updated at each step. Additionally, I added an additional convolutional layer, as well as using a max pooling layer on each of these. Thus our current network architecture is:

```

class MyNetwork(nn.Module):

    def __init__(self):
        super(MyNetwork, self).__init__()
        self.conv1 = nn.Conv2d(3,6,3)
        self.pool1 = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6,16,3)
        self.pool2 = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(in_features=576, out_features=512)
        self.fc2 = nn.Linear(in_features=512, out_features=100)

    def forward(self, x):
        x = self.pool1(F.relu(self.conv1(x)))
        x = self.pool2(F.relu(self.conv2(x)))
        x = x.view(x.size(0), -1) # flatten
        x = self.fc1(x)
        x = self.fc2(x)
        return x

```

Surprisingly, this gave much worse results than anything before, accuracy not even climbing past 2%. However, as the accuracy had not seemed to level off as much as prior examples,

I increased the number of epochs from 10 to 25, giving the network more opportunities to pick up features from the training data. This revealed something very remarkable—the best improvements to the network only started to be experienced after 10 epochs. Not seeing any signs of overfitting, I further increased the number of epochs to 50, and was able to see further improvements still.

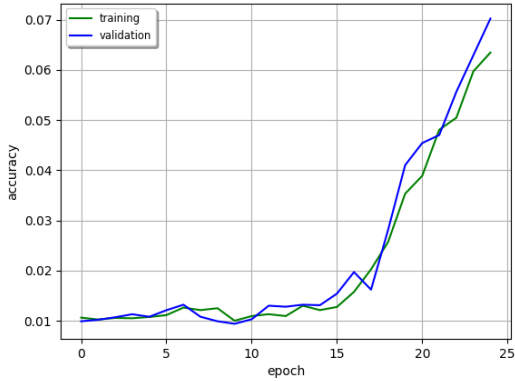


Figure 3: 25 epochs, accuracy still rapidly climbing—notice no improvements until after 10 epochs

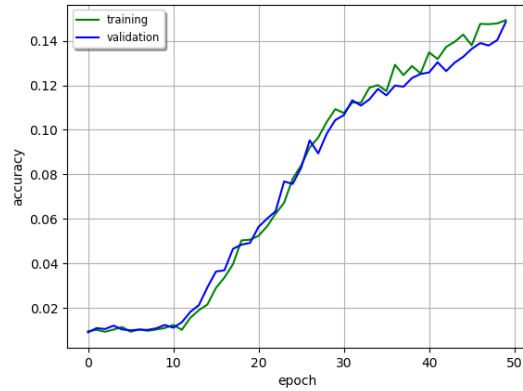


Figure 4: 50 epochs, still no signs of overfitting

2 Generating a Pegasus