Ana Estrada

1. When calculating the gradient descent of the learning algorithm the momentum is multiplied by the change in weight for a given layer and is added to the learning rate multiplied by the change in weights. Momentum can accelerate the training and testing schedule. It does this by accelerating the gradient descent in the correct direction. It tells the learning rate if it is strong enough to skip over potentially larger local minima. Momentum works like momentum in physics. It basically will either allow the learning algorithm to make it out of a valley or descent into a valley based on the speed provided by the momentum. You have to be careful when setting the value because if the number is too high then the learning algorithm might skip over potentially lower points on the cost function. It performs well on small or noisy gradients.
2. It initializes all the weights to numbers within a normal distribution. It takes into account the nonlinearity of the activation functions in this case the elu function. According to the tensor flow documentation it creates a truncated normal distribution centered around 0 based on the number of input the weight matrix is taking in. Initializing the weights to small meaningful numbers is to ensure that we do not have a dominating signal and we really do not want them to be 0 because if they were 0 then the dot product with the weight matrix wouldn't give us a signal. Avoiding 0 and large numbers makes sure that we can actually tune the weights in a meaningful way.
3. The elu activation is similar in benefits to the Relu activation function in that it does not fall prey to the vanishing gradient in the same way that the Sigmoid activation function does. Elu also does not output signals of 0 so it avoids the dying Relu function. The dying Relu problem is that the Relu function will output a 0 activation at some point and this causes neurons to die so we are not getting information from these neurons and that could be messing up our model, but the elu activation function gets around this problem because it does not output 0. It is also differentiable at all points due to the shape of the graph unlike the relu activation function.
4. Is a technique used when your data belongs exclusively to one category which makes sense in this case because clothing can only be of one type normally. This is what the categorical part means. The sparse part means that the classifications are not one hot encoded instead they are encoded with integers. We have to use this because we have a softmax layer at the end of our model.
5. Batch normalization is a technique that aims to create a faster and better convergence by normalizing input layers and preventing changes in their distribution during training. Helps us avoid vanishing and exploding gradients. It works by normalizing each layer input with the mini-batch's mean and standard deviation. It zero-centers and normalizes each input and then scales and shifts the result using two new parameter vectors per layer. One parameter layer for scale and one for shifting. The model will learn the optimal scale and mean for each of the layer's inputs. This can replace standardizing the input features and can be a regularization technique. It is a way to ensure that we have a balanced

learning process and weights are not disappearing nor are some becoming more dominant than they need to be.

6. The accuracy of my model was 87.62% for the validation and the loss was 0.3598 for the validation.

Sources

The book, Professor Watson, https://arxiv.org/abs/1903.06733, https://titanwolf.org/Network/Articles/Article?AID=6391583a-fda4-439d-a7f0-2dd1202d0a0f#gsc.tab=0, I discussed the order of operation with Danseh