Neural Networks and their Applications

2022 Spring

Assignment 2

2002387 Mahmoud Aboud Nada *Instructor: Dr. Hazem Abbas*

§ Home Work 2 §

Problem 1:

(1) an image of shape 500x500x3 has the shape of 750,000 after flattening, when it passes through fully connected layer with 100 hidden unites the output shape becomes 100x750,000 = 750,000,000.

(2) $parameters = C_{out} * C_{in} * K * K + bais = 3 * 10 * 5 * 5 + 0 = 750$

(3) First image used a Vertical line detector filter which holds the values $\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$

Second image used a horizontal line detector which holds the values $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

(4)

(5) $\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$

- Improves gradient flow through the network
- Allows higher learning rates
- Reduces the strong dependence on initialization
- Acts as a form of regularization

(6)

(7)

(8) Inverted dropout is a variant of the original dropout technique developed by Hinton et al. Just like traditional dropout, inverted dropout randomly keeps some weights and sets others to zero. This is known as the "keep probability" The one difference is that, during the training of a neural network, inverted dropout scales the activations by the inverse of the keep probability $q = 1 - p$.
This prevents the network's activations from getting too large and does not require any changes to the network during evaluation. In contrast, traditional dropout requires scaling to be implemented during the test phase.

(9) fully connected layers work better with one-dimensional data because it needs its input to be flattened, this eliminates the features that depend on certain positions in images.

(10)

(11) Reduce the learning rate at a few fixed points. E.g. for ResNets, multiply LR by 0.1 after epochs 25, 60.

(12) Convolutions are not densely connected, not all input nodes affect all output nodes. This gives convolutional layers more flexibility in learning. Moreover, the number of weights per layer is a lot smaller, which helps a lot with high-dimensional inputs such as image data.

(13) During training time, dropout randomly sets node values to zero. In the original implementation, we have "keep probability" p. So dropout randomly kills node values with "dropout probability" 1-p. During inference time, dropout does not kill node values, but all the weights in the layer were multiplied by p. One of the major motivations of doing so is to make sure that the distribution of the values after affine transformation during inference time is close to that during training time. Equivalently, This multiplier could be placed on the input values rather than the weights.

(14) The distinction between Momentum method and Nesterov Accelerated Gradient updates was shown by Sutskever et al, i.e., both methods are distinct only when the learning rate is reasonably large. When the learning rate is relatively large, Nesterov Accelerated Gradients allows larger decay rate than Momentum method, while preventing oscillations. The theorem also shows that both Momentum method and Nesterov Accelerated Gradient become equivalent when is small.

(15) Decay the learning rate for parameters in proportion to their update history (more updates means more decay).