भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Assignment - 3

Vinta Reethu - ES18BTECH11028

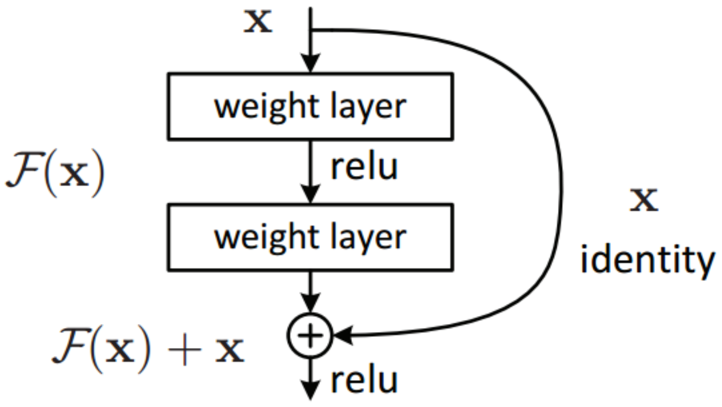
March 11, 2021

1 Feedforward and Backprop in ResNets

In the feedforward network,

- Input to the ResNet block is x
- Output of the first layer is z_1.
- Output of second layer is z_2.
- The final output of ResNet block is z_3 i.e $z_3 = z_2 + x$

A basic residual block of ResNet is given below.


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

From above we can say that

$$\mathcal{F}(x) = z_2$$
$$y = ReLU(\mathcal{F}(x) + x)$$
$$y = ReLU(z_3)$$
$$\frac{\partial z_3}{\partial z_2} = 1$$
$$\frac{\partial z_3}{\partial x} = \frac{\partial z_2}{\partial x} + 1$$

For backpropogation let's begin from the last layer,

$$\begin{aligned}\frac{\partial L}{\partial z_3} &= \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_3} \\ &= \frac{\partial L}{\partial y} \times (z_3 > 0)\end{aligned}$$

Let's find the gradient wrt z2

$$\begin{aligned}\frac{\partial L}{\partial z_2} &= \frac{\partial L}{\partial z_3}\frac{\partial z_3}{\partial z_2} \\ &= \frac{\partial L}{\partial y} \times (z_3 > 0)\end{aligned}$$

Gradient wrt z1

$$\begin{aligned}\frac{\partial L}{\partial z_1} &= \frac{\partial L}{\partial z_2}\frac{\partial z_2}{\partial z_1} \\ &= \frac{\partial L}{\partial y} \times (z_3 > 0) \times \frac{\partial z_2}{\partial z_1}\end{aligned}$$

Gradient wrt x

$$\begin{aligned}\frac{\partial L}{\partial x} &= \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial x} \\ &= \frac{\partial L}{\partial y} \times \frac{\partial y}{\partial z_3} \times \frac{\partial z_3}{\partial x} \\ &= \frac{\partial L}{\partial y} \times (z_3 > 0) \times (1 + \frac{\partial z_2}{\partial x}) \\ &= \frac{\partial L}{\partial y} \times (z_3 > 0) \times (1 + \frac{\partial z_2}{\partial z_1}\frac{\partial z_1}{\partial x})\end{aligned}$$

2 Convolutional Neural network

Given that there are 4 consecutive 3*3 convolutional layers with stride 1 and no pooling.
Let us consider a neuron in the 4th layer of size 1*1.

- Let the size of this third layer be n. The relation between the size of third layer and this 1*1 neuron(part of 4th layer) is given by

$$\begin{aligned}\lfloor \frac{n - f + 2p}{s} \rfloor + 1 &= 1 \\ \lfloor \frac{n - 3 + 0}{1} \rfloor + 1 &= 1 \\ n &= 3\end{aligned}$$

- Let the size of this second layer be n. The relation between the size of second layer and third layer is given by

$$\begin{aligned}\lfloor \frac{n - f + 2p}{s} \rfloor + 1 &= 3 \\ \lfloor \frac{n - 3 + 0}{1} \rfloor + 1 &= 3 \\ n &= 5\end{aligned}$$

- Let the size of the first layer be n. The relation between the size of first layer and second layer is given by

$$\begin{aligned}\lfloor\frac{n-f+2p}{s}\rfloor+1&=5\\ \lfloor\frac{n-3+0}{1}\rfloor+1&=5\\ n&=7\end{aligned}$$

- Let the size of the image be n. The relation between the size of image and first layer is given by

$$\begin{aligned}\lfloor\frac{n-f+2p}{s}\rfloor+1&=7\\ \lfloor\frac{n-3+0}{1}\rfloor+1&=7\\ n&=9\end{aligned}$$

  Hence, the support of a neuron in the 4th non-image layer of this network is $9\times 9=\mathbf{81}$ pixels.

3 Bias and Variance of learned model

- If the number of hidden layers are increased then the model would be able to learn more complex functions, but it won't be able to generalize to the new unseen data. Hence it **overfits** the model on the training data.
- As a result it decreases the bias and increases the variance.

4 Activation Functions

Consider a 2 layer neural network as :

$$y_k(\mathbf{x},\mathbf{w})=\sigma\left(\sum_{j=1}^{M}w_{kj}^{(2)}h\left(\sum_{i=1}^{D}w_{ji}^{(1)}x_i+w_{j0}^{(1)}\right)+w_{k0}^{(2)}\right)$$

Given that there are two activation function Tanh and Sigmoidh. Let us see if there is a relation between these two functions.

$$\sigma(a)=\frac{1}{1+\exp(-a)}$$

$$\begin{aligned}
\tanh(a) &= \frac{e^a - e^{-a}}{e^a + e^{-a}} \\
\tanh(a) + 1 &= \frac{e^a - e^{-a}}{e^a + e^{-a}} + \frac{e^a + e^{-a}}{e^a + e^{-a}} \\
&= \frac{2e^a}{e^a + e^{-a}} \\
&= \frac{2}{1 + e^{-2a}} \\
&= 2\sigma(2a) \\
\tanh(a) &= 2\sigma(2a) - 1
\end{aligned}$$

Let the parameters of the network with tanh be w^{1t} and w^{2t} and the parameters for sigmoid be w^{1s} and w^{2s}.
The output of network with sigmoid in last layer is

$$a_i^s = \sum_{j=1}^{M} w_{ij}^{2s} \sigma\left(a_j^s\right) + w_{i0}^{2s}$$

The output of network with tanh in last layer is

$$\begin{aligned}
a_i^t &= \sum_{j=1}^{M} w_{ij}^{2t} \tanh\left(a_j^t\right) + w_{i0}^{2t} \\
&= \sum_{j=1}^{M} w_{ij}^{2t} \left(2\sigma(2a_j^t) - 1\right) + w_{i0}^{2t} \\
&= \sum_{j=1}^{M} 2w_{ij}^{2t} \sigma(2a_j^t) - \sum_{j=1}^{M} w_{ij}^{2t} + w_{i0}^{2t}
\end{aligned}$$

The above two networks will be equivalent when

$$\sum_{j=1}^{M} 2w_{ij}^{2t} \sigma(2a_j^t) - \sum_{j=1}^{M} w_{ij}^{2t} + w_{i0}^{2t} = \sum_{j=1}^{M} w_{ij}^{2s} \sigma\left(a_j^s\right) + w_{i0}^{2s}$$

The above equality holds when

$$\begin{aligned}
w_{i0}^{2s} &= -\sum_{j=1}^{M} w_{ij}^{2t} + w_{i0}^{2t} \\
w_{ij}^{2s} &= 2w_{ij}^{2t} \\
a_j^s &= 2a_j^t
\end{aligned}$$

The above three equations can easily be obtained by tweaking the parameters w^{1t},w^{2t},w^{1s} and w^{2s}. Therefore she can derive an equivalent network which computes the same function.

5 Quadratic error function

Consider a quadratic error defined by

$$E(\mathbf{w}) = E\left(\mathbf{w}^{\star}\right) + \frac{1}{2}\left(\mathbf{w} - \mathbf{w}^{\star}\right)^{\mathrm{T}} \mathbf{H}\left(\mathbf{w} - \mathbf{w}^{\star}\right) \tag{1}$$

Here H is the Hessian matrix which is evaluated at $\mathbf{w}^{\star}$. Let this eigenvalue equation be

$$\mathbf{H}\mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{2}$$

where the eigenvectors $\mathbf{u}_i$ form a complete orthonormal set so that

$$\mathbf{u}_i^{\mathrm{T}} \mathbf{u}_j = \delta_{ij} \tag{3}$$

We now expand $(\mathbf{w} - \mathbf{w}^{\star})$ as a linear combination of the eigenvectors into the form

$$\mathbf{w} - \mathbf{w}^{\star} = \sum_i \alpha_i \mathbf{u}_i \tag{4}$$

Substituting Eq 4 into Eq 1, and using Eq 2 and Eq 3, the error function can be written in the form

$$E(\mathbf{w}) = E\left(\mathbf{w}^{\star}\right) + \frac{1}{2}\sum_i \lambda_i \alpha_i^2$$

Let us obtain the contour by setting the above error function to C.

$$E(\mathbf{w}) = E\left(\mathbf{w}^{\star}\right) + \frac{1}{2}\sum_i \lambda_i \alpha_i^2 = C$$

In the above equation C,$E(\mathbf{w}^{\star})$ are constant. Hence we can obtain that

$$\sum_i \lambda_i \alpha_i^2 = B$$

where B is a constant whose value is $2C - 2E\left(\mathbf{w}^{*}\right)$

Hence we can say that the contours of constant error are ellipses whose axes are aligned with the eigenvector $\mathbf{u_i}$ of the Hessian Matrix H. In order to obtain the length of j th axis let us set $\alpha_i = 0$ and i $\neq j$. Therefore we will obtain that

$$\alpha_j = \sqrt{\frac{B}{\lambda_j}}$$

lengths that are inversely proportional to the square root of the corresponding eigenvalues $\lambda_{\mathbf{j}}$.

6 Transfer Learning

- It is given that Kaziranga National Park has collected 20 images of 200 species of animal.
- They have access to a deep learning model deployed at Olympic National Park, Washington trained on 1 million images from 1000 classes.
- Since both the target and source datasets are **similar** the important features will also be same.
- They just have to remove the output layer and add 200 neurons and train only this output layer by freezing the weights of the the remaining layers.
- In this way they don't have to waste time by writing new network, training and hyper parameter tuning.
- This is the best possible way as, if we were to train a new neural network with these 4000 image it would overfit as the number of data points are very less.