Constraining MLP

Suppose the input is a 2-dimensional tensor $\mathbf{X}$ representing an image, and the immediate hidden layer is also a 2-dimensional tensor $\mathbf{H}$ having the same shape as $\mathbf{X}$, and each element of $\mathbf{H}$ is generated by all pixels of $\mathbf{X}$, then for each element $[\mathbf{H}]_{i,j}$, we need a 2-dimentiol weight metrix $\mathbf{W_{i,j}}$, and each $\mathbf{W_{i,j}}$ has the same shape as $\mathbf{X}$. So in all we need a 4th-ordered weight tensor W.

In mathematics, suppose $\mathbf{U}$ contains biases, we can generate $\mathbf{H}$ by:

$$[\mathbf{H}]_{i,j} = [\mathbf{U}]_{i,j} + \sum_k \sum_l [\mathsf{W}]_{i,j,k,l} [\mathbf{X}]_{i,j}$$

If we do some modifications on the above formula, let $k = i + a$ and $l = j + b$ where a and b are offsets that can be both positive and negtive and 0, covering the whole image:

$$[\mathbf{U}]_{i,j} + \sum_a \sum_b [\mathsf{V}]_{i,j,a,b} [\mathbf{X}]_{i+a,j+b}$$

The above formula is like summing over weighted pixels around $[\mathbf{X}]_{i,j}$.

translation invariance

For object recognition task, if we shift an object to another position, we should still be able recognize it. It means, for a CNN doing object recognition, the hidden layers should not rely on its location.

So for the above formula, V and $\mathbf{U}$ do not actually depend on (i,j), this only happens when $[\mathsf{V}]_{i,j,a,b} = [\mathbf{V}]_{a,b}$ and $\mathbf{U}$ is a constant. Thus we can simplify the definition of $\mathbf{H}$:

$$[\mathbf{H}]_{i,j} = u + \sum_a \sum_b [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}.$$

And this is convolution. We shrink the scale of $\mathbf{V}$ by 2 order.

locality

We should not go to far away from $[\mathbf{X}]_{i,j}$ to assess waht is going on at $[\mathbf{H}]_{i,j}$, which means a and b should have a limited range.

So we can rewrite the definition of $[\mathbf{H}]_{i,j}$ as

$$[\mathbf{H}]_{i,j} = u + \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}$$

And this is a convolutional layer. $\mathbf{V}$ is the convolution kernel.