Deep Learning

Multilayer Perceptron (MLP)

- MLP: Stacking linear layer and nonlinear activations.
- Through many non-linear layers, transform a linear nonseparable problem to linear separable at the last layer

NOTE: Only stacking linear layers will not get a linear model. Only stacking nonlinear layers will not be able to modify the dimension of the input, and the output will be plain in some cases(such as Relu activation function, only the first activation is functioning).

upgrade parameters - stochatistic gradient descent (SGD)

- SGD: update the parameters of the model by taking a step in the direction determined by a randomly chosen mini-batch of training examples. It can hopefully escape from local minima and converge to the global minimum.

Activation Function

- Sigmoid: S-shaped curve, output range (0,1)
- Tanh: S-shaped curve, output range (-1,1)
- ReLU: Rectified Linear Unit, output range (0,infinity)
- Leaky ReLU: Leaky version of ReLU, output range (-infinity,infinity)

NOTE:

- ReLU is the most commonly used activation function. It is locally linear, so the seperation curve that ReLU forms is a plane locally.
- Leaku ReLU is usually used in tricky training scenarios, like GAN
- sigmoid and tanh suffer from saturation zone.

Problem of Using MLP

- Flatten an image into a vector would be very expensive for high resolution images
- Flattening operation breaks the local structure of an image.

Convolutional Neural Network (CNN)

- CNN: stack CONV, POOL, and FC layers
- Trend towards smaller filters and deeper networks
- Trend towards getting rid of POOL/FC layers(just CONV)
- historical architecture:

$$[(CONV - RELU) * N - POOL] * M - (FC - RELU) * K, SOFTMAX$$

  where N is usually up to ~5, M is large, 0<=k<=2
  - recent advances such as ResNet/GoogLeNet changed this parradigm

NOTE: a CONV layer may has a bias. For a single filter, only one bias is needed. WHY? Because different bias will break the translation equivariance of the CONV.

Pooling Layer

- max pooling: take the maximum value in a window
- average pooling: take the average value in a window

NOTE: pooling layer gives CNN rotation invariance. When we need to detect something, max pooling should be used, it mean: "I don't care where the object is, as long as it exists". When we just need a semantic understanding, average pooling is ok.

Comparision between FC an CONV layer

Assume input size: $W_1 \times H_1 \times C$, output size: $W_2 \times H_2 \times K$

- FC
  - Densely connected
  - Total Parameter: $W_1 W_2 H_1 H_2 C K$
- CONV
  - Filter size: F
  - Total Parameter: $F^2 C K$

why does CNN has fewer parameters than MLP?

1. Sparse Connectivity: Convolutional networks typically have sparse interactions (also referred to as sparse connectivity or sparse weights). This is accomplished by making the kernel smaller than the input.
2. Parameter Sharing: Convolutional networks share parameters across the entire input(slide the same kernel over the entire input).

which one is better

1. FC is more expressive! FC is a super set of CNN (without sparse and parameter sharing constraints.)
2. FC is susceptible to translation and rotation
3. Conv is equivariant with translation, and applying pooling induces invariance to small translations and rotations.
4. Thus CNN is better for training.