Practical Lecture 7 - Convolution Networks, Recurrent Networks

Machine Learning Course - 2nd Semester 2019/2020

Instituto Superior Tecnico, Universidade de Lisboa

1 Convolution Networks

1) a) For a kernel of shape $(k_h, k_w) = (5, 3)$ with one channel, a stride of $(s_h, s_w) = (1, 1)$ and an input of shape $(H, W, C) = (6, 5)$ with one channel, what padding (p_h, p_w) is required such that the output has the same shape as the input (also called same padding)?

b) For a general kernel shape (k_h, k_w), a general input shape (H, W) and a fixed stride of $(s_h, s_w) = (1, 1)$. Can you derive a general formula for (p_h, p_w) such that the output has shape (H, W)?

2) Connsider the following network:

- Input:
  - $(H, W, C) = (28, 28, 3)$
- Convolution:
  - Number of kernels: 8
  - For all kernels the shapes are: $(k_h, k_w, k_c) = (5, 5, 3)$
  - The strides are: $(s_h, s_w) = (1, 1)$
  - Same padding
- Pooling:
  - $(k_h, k_w) = (4, 4)$
  - $(s_h, s_w) = (2, 2)$

a) List and count all parameteres in this network.

b) What is the output's shape?

3) Connsider the following network:

- Input:
  - $(H, W, C) = (10, 10, 1)$
- Convolution:
  - Number of channels: 8
  - $(k_h, k_w, k_c) = (5, 5, 1)$
  - $(s_h, s_w) = (1, 1)$
  - Same padding
- Convolution:
  - Number of channels: 12

- $(k_h, k_w, k_c) = (3, 3, 8)$
- $(s_h, s_w) = (1, 1)$
- Same padding

a) List and count all parameteres in this network.
b) What is the output's shape?

4) Given an input image of shape $(10, 10, 1)$ connected to a layer of $(10, 10, 5)$ units.

a) How many parameters does the network have if it is fully connected?

b) What about if it is a convolution layer with 5 kernels of shape $(3, 3, 1)$ with same padding and unitary stride.

5) For the following input image:

$$I = \begin{pmatrix} 10 & 10 & 10 & 10 & 10 & 10 \\ 5 & 5 & 5 & 5 & 5 & 5 \\ 2 & 2 & 2 & 0 & 0 & 2 \\ 2 & 2 & 2 & 0 & 0 & 2 \\ 50 & 10 & 50 & 50 & 50 & 10 \\ 50 & 10 & 50 & 50 & 50 & 10 \end{pmatrix}$$

a) What is the output provided by a convolution layer with the following properties:

- Zero Padding: $(1, 1)$
- Stride: $(1, 1)$
- Kernel Shape: $(3, 3)$
- Number of kernels: 2
- Kernels:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

b) Take the output from **a)** and apply a max pooling layer with the following properties:

- Zero Padding: $(0, 0)$
- Stride: $(2, 2)$
- Kernel Shape: $(2, 2)$

6) For the following kernels, describe what kind of feature they extract from the image:

$$F_1 = \begin{pmatrix} -10 & -10 & -10 \\ 5 & 5 & 5 \\ -50 & -10 & -10 \end{pmatrix}, F_2 = \begin{pmatrix} 2 & 2 & 2 \\ 2 & -12 & 2 \\ 2 & 2 & 2 \end{pmatrix}, F_3 = \begin{pmatrix} -20 & -10 & 0 & 5 & 10 \\ -10 & 0 & 5 & 10 & 5 \\ 0 & 5 & 10 & 5 & 0 \\ 5 & 10 & 5 & 0 & -10 \\ 10 & 5 & 0 & -10 & -20 \end{pmatrix}$$

2 Recurrent Networks

1) Consider a recurrent network with one hidden layer to solve a **many-to-many** task. Take into account that:

- The recurrent connections happen between the hidden layers
- The hidden activation function is the hyperbolic tangent
- The output activation function is softmax
- The error function is the cross-entropy between output and target

a) Write down the model equations for forward propagation.
b) Write down the model equations for backward propagation.
c) Perform a stochastic gradient descent update for:

$$\mathbf{x}^{(1)} = \left[\begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 8 \\ 2 \\ 0 \end{pmatrix} \right]$$

With targets:

$$t^{(1)} = \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

Initialize all weights and biases to 0.1, using 3 units per hidden layer, initializing the hidden state to all zeros and using $\eta = 1.0$.

2) Consider a recurrent network with one hidden layer to solve a **many-to-many** task. Take into account that:

- The recurrent connections happen between the hidden layers
- The hidden activation function is the hyperbolic tangent
- The output activation function is the identity
- The error function is the half squared error between output and target

a) Write down the model equations for forward propagation.
b) Write down the model equations for backward propagation.
c) Perform a stochastic gradient descent update for:

$$x^{(1)} = \left[\begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 8 \\ 2 \\ 0 \end{pmatrix} \right]$$

With targets:

$$t^{(1)} = \left[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]$$

Initialize all weights and biases to 0.1, using 3 units per hidden layer, initializing the hidden state to all zeros and using $\eta = 1.0$.

3) Consider a recurrent network with one hidden layer to solve a **many-to-one** task. Take into account that:

- The recurrent connections happen between the hidden layers
- The hidden activation function is the hyperbolic tangent
- The output activation function is the softmax
- The error function is the cross-entropy between output and target

a) Write down the model equations for forward propagation.
b) Write down the model equations for backward propagation.
c) Perform a stochastic gradient descent update for:

$$x^{(1)} = \left[\begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 8 \\ 2 \\ 0 \end{pmatrix} \right]$$

With targets:

$$\mathbf{t} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Initialize all weights and biases to 0.1, using 3 units per hidden layer, initializing the hidden state to all zeros and using $\eta = 1.0$.

4) Consider a recurrent network with one hidden layer to solve a **many-to-one** task. Take into account that:

- The recurrent connections happen between the hidden layers
- The hidden activation function is the hyperbolic tangent
- The output activation function is the identity
- The error function is the half squared error between output and target

a) Write down the model equations for forward propagation.
b) Write down the model equations for backward propagation.
c) Perform a stochastic gradient descent update for:

$$x^{(1)} = \left[\begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 8 \\ 2 \\ 0 \end{pmatrix} \right]$$

With targets:

$$\mathbf{t} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Initialize all weights and biases to 0.1, using 3 units per hidden layer, initializing the hidden state to all zeros and using $\eta = 1.0$.

3 Thinking Questions

a) Why is it said that backpropagation through time is very hard to implement in parallel?

b) What factors do you think play a role when choosing a CNNs hyperparameters?

c) Can you see why is it said that a CNN is a smart way to regularize an MLP through a different architecture? Think about some key factors like free parameters, bias and generalization.