# CS280 Fall 2018 Assignment 2
# Part A

CNNs

Due in class, Nov 02, 2018

**Name: Ke Zhang**

**Student ID: 50369264**

# 1. Linear Regression(10 points)

- Linear regression has the form $E[y|x] = w_0 + \boldsymbol{w^T}x$. It is possible to solve for $\boldsymbol{w}$ and $w_0$ seperately. Show that

$$w_0 = \frac{1}{n}\sum_i y_i - \frac{1}{n}\sum_i x_i^T\boldsymbol{w} = \overline{y} - \overline{x}^T\boldsymbol{w}$$

- Show how to cast the problem of linear regression with respect to the absolute value loss function, $l(h, x, y) = |h(x) - y|$, as a linear program.

**Solution**

- The linear regression equation can be rewritten as

$$y = w_0 + \boldsymbol{w}^T x + \epsilon$$

where $\epsilon$ is an error singal with zero mean $E(\epsilon) = 0$. We have known that the estimator of these weights(i.e., the solution of maximum likelihood estimation) is solved from least square equation. Hence, to compute $w_0$:

$$\begin{aligned} l(w) &= \frac{1}{2n}\sum_i (y_i - (w_0 + \boldsymbol{w}^T x_i))^2 \\ \frac{\partial L(w)}{\partial w_0} &= \frac{1}{n}\sum_i (y_i - (w_0 + \boldsymbol{w}^T x_i)) \\ 0 &= \frac{1}{n}\sum_i (y_i - \boldsymbol{w}^T x_i) - w_0 \\ w_0 &= \frac{1}{n}\sum_i (y_i - \boldsymbol{w}^T x_i) \\ &= \bar{y} - \bar{x}^T\boldsymbol{w} \end{aligned}$$

- If the loss function is set as $l(w) = \frac{1}{n}\sum_i (y_i - h(x))$ where $h(x) = w_0 + \boldsymbol{w}^T x$, the derivate of $w_0$ is

$$\frac{\partial l(w)}{\partial w_0} = -\frac{1}{n}\sum_i \mathrm{sgn}(y_i - h(x_i))$$

where

$$\mathrm{sgn}(y_i - h(x_i)) = \begin{cases} 1 & y_i > h(x_i) \\ -1 & y_i < h(x_i) \\ [-1, 1] & y_i = h(x_i) \end{cases}$$

so we can see that there is no explict estimator of optimal weight $w_0$ since $l_1$ norm is not smooth in each point, but the best weights could be computed interatively from the initial weights.

# 2. Convolution Layers (5 points)

We have a video sequence and we would like to design a 3D convolutional neural network to recognize events in the video. The frame size is 32x32 and each video has 30 frames. Let's consider the first convolutional layer.

- We use a set of $5 \times 5 \times 5$ convolutional kernels. Assume we have 64 kernels and apply stride 2 in spatial domain and 4 in temporal domain, what is the size of output feature map? Use proper padding if needed and clarify your notation.
- We want to keep the resolution of the feature map and decide to use the dilated convolution. Assume we have one kernel only with size $7 \times 7 \times 5$ and apply a dilated convolution of rate 3. What is the size of the output feature map? What are the downsampling and upsampling strides if you want to compute the same-sized feature map without using dilation?

Note: You need to write down the derivation of your results.

**Solution**

- Because of the size of kernel, we need to extract 5 continuous frames once to make convolution with a kernel, so the number of featurers map is:

  $$((30 - 5 + 3)/4 + 1) \times 64 = 8 \times 64$$

  here the padding is 3.
  The size of one feature map is:

  $$(32 - 5 + 1)/2 + 1 = 15$$

  Since the kernel is $5 \times 5 \times 5$, padding in one frame is set as 1, thus the size of output feature map is $8 \times 64@15 \times 15$.

- The dilated convolution could expand kernel's receptive field without changing the size of the feature map. Therefore, the output feature map is still $32 \times 32 \times 30$.
  Given the kernel with size $7 \times 7 \times 5$, actual receptive field is $19 \times 19 \times 13((7 - 1) \times 3 + 1$ and $(5 - 1) \times 3 + 1)$, so in order to keep the size of one feature map still be $32 \times 32$, padding $p$ should equal to: $p = \dfrac{19 - 1}{2} = 8$
  Without dilation, according to the kernel cascading fomula, the receptive field:

  $$r_n = r_{n-1} \cdot k_{n-1} - (r_{n-1} - \prod_{i=0}^{n-1} S_i) \cdot (k_n - 1)$$

  where $r_{n-1}$ is the size of receptive field and $k_{n-1}$ is the kernel size using at $n - 1_{th}$ cascade kernel. We assume that $r_0 = 1$ and $S_0 = 1$ at the original output layer.
  According this rule, to keep the size of receptive field still be $19 \times 19 \times 13$, suppose we need three cascading kernels with all size $7 \times 7$(here we first consider only kernel's the height and width), then we have:

  $$r_0 = 1$$

  $$r_1 = r_0 + (k_n - 1)S_0$$

  $$r_2 = r_1 + (k_n - 1)S_1$$

$$r_3 = r_2 + (k_n - 1)S_2S_1$$

then when we use **two** cascading kernel with all stride $S = 2$, for height and width: $r_2 = 7 + 6 \times 2 = 19$, for spatial size: $r_2 = 5 + 4 \times 2 = 13$;
or use **three** cascading kernel with all stride $S = 1$, for height and width: $r_3 = 7+6+6\times1 = 19$, for spatial size: $r_3 = 5 + 4 + 4 \times 1 = 13$.
And we can see that such cascading method could be used in both downsampling adn upsampling without changing the resolution feature map, hence, strides could be 2 or 1.

# 3. Batch Normalization (5 points)

With Batch Normalization (BN), show that backpropagation through a layer is unaffected by the scale of its parameters.

- Show that
$$BN(\mathbf{Wu}) = BN((a\mathbf{W})\mathbf{u})$$
where $\mathbf{u}$ is the input vector and $\mathbf{W}$ is the weight matrix, $a$ is a scalar.

- (Bonus: 5 pts) Show that
$$\frac{\partial BN((a\mathbf{W})\mathbf{u})}{\partial \mathbf{u}} = \frac{\partial BN(\mathbf{Wu})}{\partial \mathbf{u}}$$

**Solution**

-
$$BN(\mathbf{Wu}) = \frac{\mathbf{Wu} - E[\mathbf{Wu}]}{\sqrt{Var[\mathbf{Wu}]}}$$

$$\begin{aligned}
BN((a\mathbf{W})\mathbf{u}) &= \frac{(a\mathbf{W})\mathbf{u} - E[(a\mathbf{W})\mathbf{u}]}{\sqrt{Var[(a\mathbf{W})\mathbf{u}]}} \\
&= \frac{a\mathbf{Wu} - aE[\mathbf{Wu}]}{\sqrt{E[(a\mathbf{Wu})^2 - (E[a\mathbf{Wu}])^2]}} \\
&= \frac{a\mathbf{Wu} - aE[\mathbf{Wu}]}{\sqrt{a^2E[(\mathbf{Wu})^2 - a^2(E[\mathbf{Wu}])^2]}} \\
&= \frac{\mathbf{Wu} - E[\mathbf{Wu}]}{\sqrt{Var[\mathbf{Wu}]}} \\
&= BN(\mathbf{Wu})
\end{aligned}$$

- Denote $\mathbf{Wu} = \mathbf{x}$, according to the definition, we have
$$E(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} x_i$$
$$Var(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(\mathbf{x}))^2$$
where $x_i$ is the $i_{th}$ output point.
$$\begin{aligned}
\frac{\partial BN(\mathbf{x})}{\partial u_i} &= \frac{\partial BN(\mathbf{x})}{\partial x_i}\mathbf{W}_i \\
&= \frac{(1-\frac{1}{N})\sqrt{Var(x_i)} - (x_i - \frac{1}{N}\sum_{k=1}^{N} x_k)\frac{1}{2\sqrt{Var(x_i)}}\frac{2}{N}\sum_{k=1}^{N}(x_k - E(\mathbf{x}))(-\frac{1}{N})}{Var(x_i)}\mathbf{W}_i \\
&= \frac{(1-\frac{1}{N})\sqrt{Var(x_i)} + (x_i - \frac{1}{N}\sum_{k=1}^{N} x_k)\frac{1}{N^2\sqrt{Var(x_i)}}\sum_{k=1}^{N}(x_k - E(\mathbf{x}))}{Var(x_i)}\mathbf{W}_i
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial BN(a\mathbf{x})}{\partial u_i} &= \frac{\partial BN(a\mathbf{x})}{\partial x_i}\mathbf{W}_i \\
&= \frac{a(1-\frac{1}{N})a\sqrt{Var(x_i)} - a(x_i - \frac{1}{N}\sum_{k=1}^{N} x_k)\frac{1}{2a\sqrt{Var(x_i)}}\frac{2}{N}\sum_{k=1}^{N} a(x_k - E(\mathbf{x}))(-\frac{a}{N})}{a^2Var(x_i)}\mathbf{W}_i \\
&= \frac{(1-\frac{1}{N})\sqrt{Var(x_i)} + (x_i - \frac{1}{N}\sum_{k=1}^{N} x_k)\frac{1}{N^2\sqrt{Var(x_i)}}\sum_{k=1}^{N}(x_k - E(\mathbf{x}))}{Var(x_i)}\mathbf{W}_i \\
&= \frac{\partial BN(\mathbf{x})}{\partial u_i}
\end{aligned}$$

therefore, $\frac{\partial BN(a\mathbf{x})}{\partial \mathbf{u}} = \frac{\partial BN(\mathbf{x})}{\partial \mathbf{u}}$.