Introduction to Machine Learning and Deep Learning

Lecture 5

TensorFlow “Hello World!” in python (using constant)

- Import TensorFlow module in python

```
import tensorflow as tf
```

- Define a constant

```
hello_c = tf.constant('Hello, TensorFlow constant!')
```

- Create TensorFlow Session

```
sess = tf.Session()
```

- Run TensorFlow Session

```
result = sess.run(hello_c)
print(result)
```

TensorFlow “Hello World!” in python (using variable)

- Import TensorFlow module in python

```
import tensorflow as tf
```

- Define a Variable

```
hello_v = tf.Variable('Hello, TensorFlow variable!')
```

- Create TensorFlow Session

```
sess = tf.Session()
```

- Initialize Variables

```
init = tf.global_variables_initializer()
sess.run(init)
```

- Run TensorFlow Session

```
result = sess.run(hello_c)
print(result)
```

http://localhost:8888/notebooks/0_HelloWorld.ipynb

Tensorflow “Hello World!” in python (Eager Execution)

- Import TensorFlow module in python

```
import tensorflow as tf
```

- Enable Eager Execution

```
tf.enable_eager_execution()
```

- Define a Variable

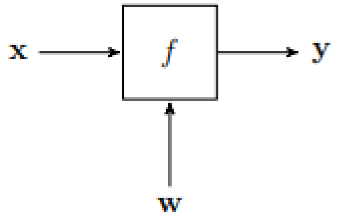
```
x = [[2.]]
```

- Run your analysis

```
m = tf.matmul(x, x) #Matrix multiplication
print("hello, {}".format(m))
```

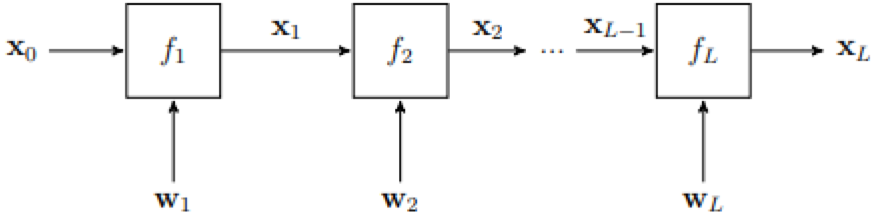
http://localhost:8888/notebooks/1_EagerExecution.ipynb

Typical CNN Architecture

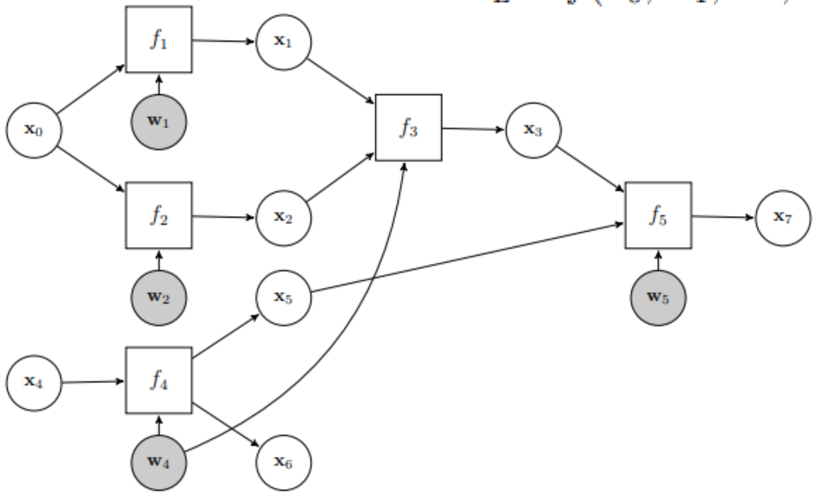


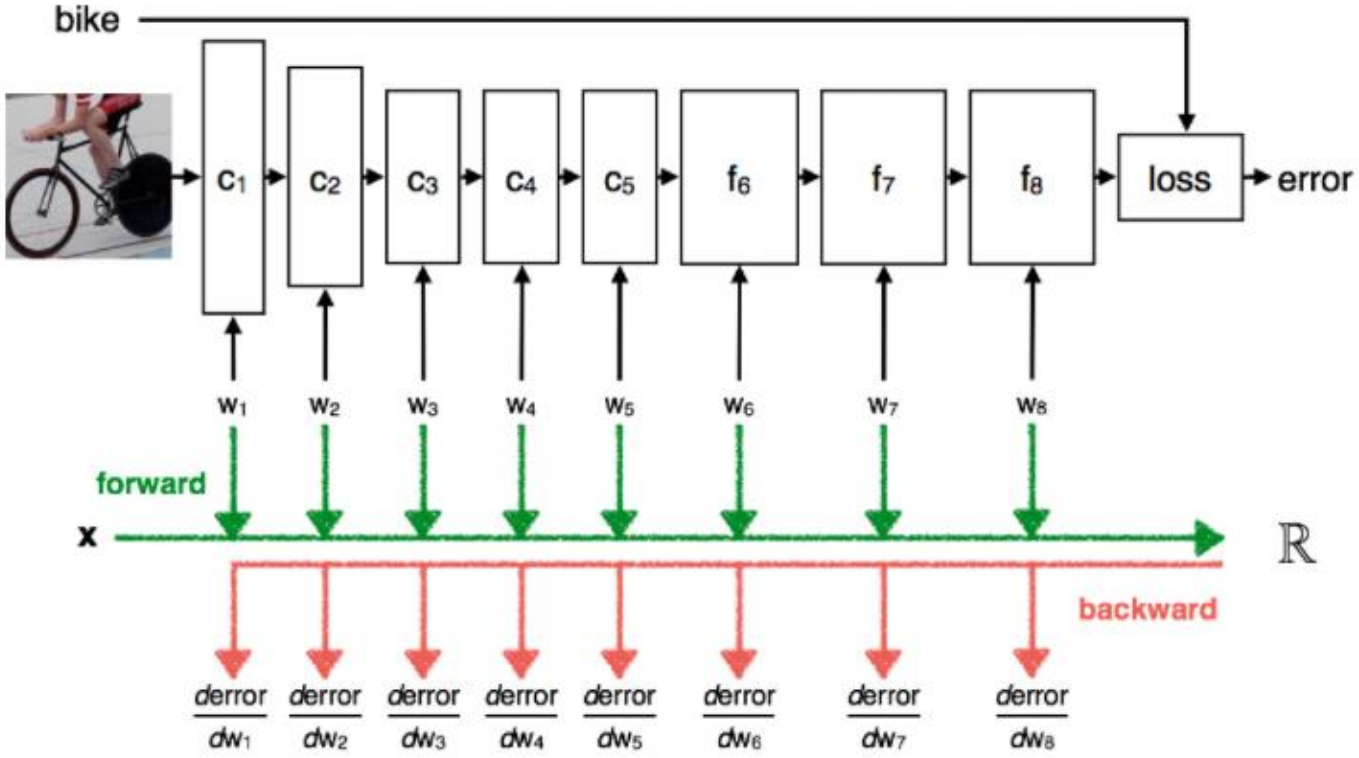
$$y = f(x; w)$$

Convolution



$$\mathbf{x}_L = f(\mathbf{x}_0; \mathbf{w}_1, \ldots, \mathbf{w}_L).$$





http://www.vlfeat.org/matconvnet/

Example 1

Problem: Given set of points {(-1.0, -1.5), (0.0, 0.0), (1.0, 1.5), (2.0, 3.0), (3.0, 4.5)}. Find the line which passes through these points.

The equation of a line can be written as
$y = f(x; w, b) = wx + b$, where ‘w’ is the slope and ‘b’ is the intercept or bias. As the line passes through (0.0, 0.0), the intercept is zero.

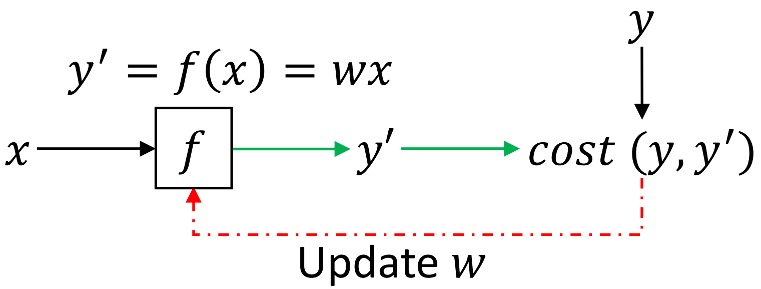
The slope w can be calculated using

$$w = \frac{y_2 - y_1}{x_2 - x_1} = \frac{1.5 - 0}{1 - 0} = \frac{1.5}{1} = 1.5$$

Equation of Line is $\boldsymbol{y = 1.5(x)}$

Solve Example 1 using Convolutional Neural Network (CNN)

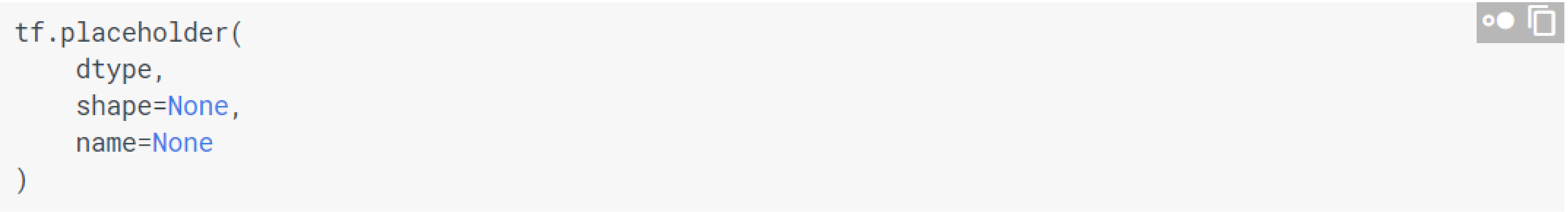
- Problem: Given set of points {(-1.0, -1.5), (0.0, 0.0), (1.0, 1.5), (2.0, 3.0), (3.0, 4.5)}. Find the line which passes through these points.
- $y = f(x; w) = wx$, where 'w' is the slope (weight)
- Design a single layer CNN with a 1×1 convolutional filter with weight 'w'.
- Define a cost function $cost = costfunction(y, y')$, in this case we use mean squared error.
- Choose an optimization function. In this case we choose Gradient descent.
- Initialize the slope/weight w to a suitable value.
- Run the network for a number of iterations ($epoch$) until we are satisfied with the number of $epochs$ or $cost$.
- Update the weight w during each $epoch$.



http://localhost:8888/notebooks/2_example1.ipynb

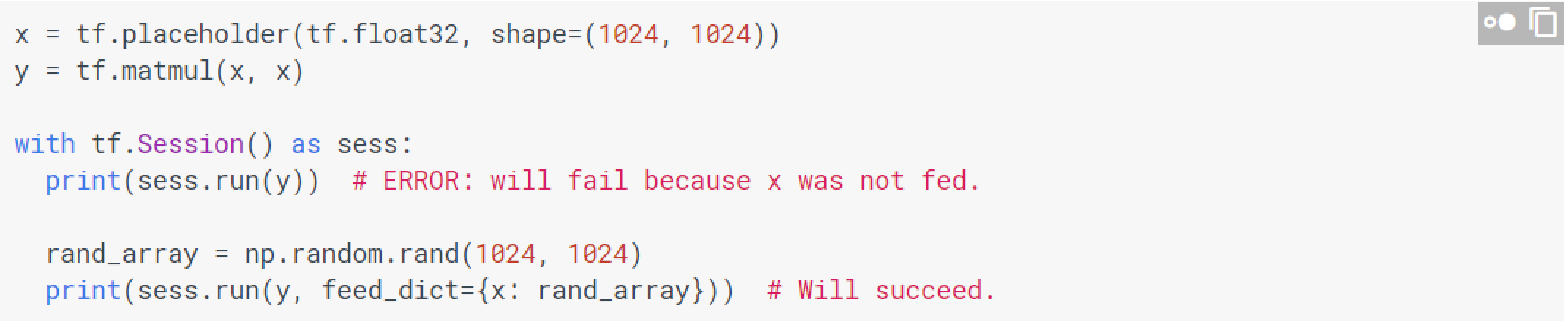
TensorFlow placeholder

Inserts a placeholder for a tensor that will be always fed.



```
tf.placeholder(
    dtype,
    shape=None,
    name=None
)
```

Example



```
x = tf.placeholder(tf.float32, shape=(1024, 1024))
y = tf.matmul(x, x)

with tf.Session() as sess:
  print(sess.run(y))  # ERROR: will fail because x was not fed.

  rand_array = np.random.rand(1024, 1024)
  print(sess.run(y, feed_dict={x: rand_array}))  # Will succeed.
```

Can be used to pass large amount of data in small batches.

Important factors in deep learning

- Data Set and ground truth Annotation
- Architecture of the network
- Initialization of parameters (weights/biases)
- Loss Function
- Number of Iterations/epochs
- Learning Rate
- Optimization Function

Example 1

Problem: Given set of points {(-1,0), (1,10), (2,15), (3,20), (4,25), (5,30)}. Find the line which passes through these points.

The equation of a line can be written as
$y = f(x; w, b) = wx + b$, where ‘w’ is the slope and ‘b’ is the intercept or
bias.

The slope w can be calculated using

$$w = \frac{y_2 - y_1}{x_2 - x_1} = \frac{10 - 0}{1 - (-1)} = \frac{10}{2} = 5$$

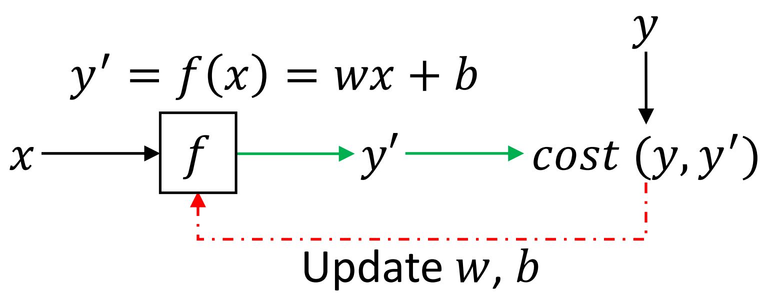
The bias b can be calculated using

$$b = y - mx = 10\ - 5\ (1) = 5$$

Equation of Line is $\boldsymbol{y = 5(x) + 5}$

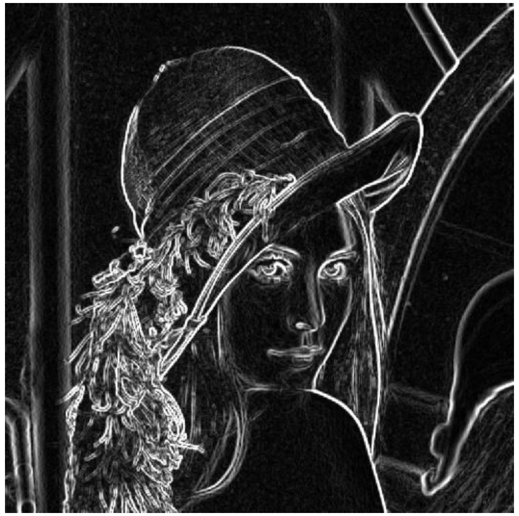
Solve Example 1 using Convolutional Neural Network (CNN)

- Problem: Given set of points {(-1,0), (1,10), (2,15), (3,20), (4,25), (5,30)}. Find the line which passes through these points.
- $y = f(x; w) = wx + b$, where ‘w’ is the slope (weight) and ‘b’ is the intercept or bias.
- Design a single layer CNN with a 1×1 convolutional filter with weight ‘w’ and added bias ‘b’.
- Define a cost function $cost = costfunction(y, y')$, in this case we use mean squared error.
- Choose an optimization function. In this case we choose Gradient descent.
- Initialize the weight w and bias b to a suitable value.
- Run the network for a number of iterations ($epoch$) until we are satisfied with the number of $epochs$ or $cost$.
- Update the weight w and bias b during each $epoch$.



Compute gradient of an image using deep learning techniques (in tensorflow)

Given image of Lena and a sketch which highlights the edges, produce a similar sketch for the cameraman image.



Inspiration from Sobel operator and design a network which computes the approximate gradient.
With A as input image matrix

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

~Horizontal derivative

~Vertical derivative

MNIST Example

- https://www.tensorflow.org/tutorials/layers
- https://github.com/tensorflow/models