Deep Learning / Machine Learning Seminar

- CNN techniques to RNN-

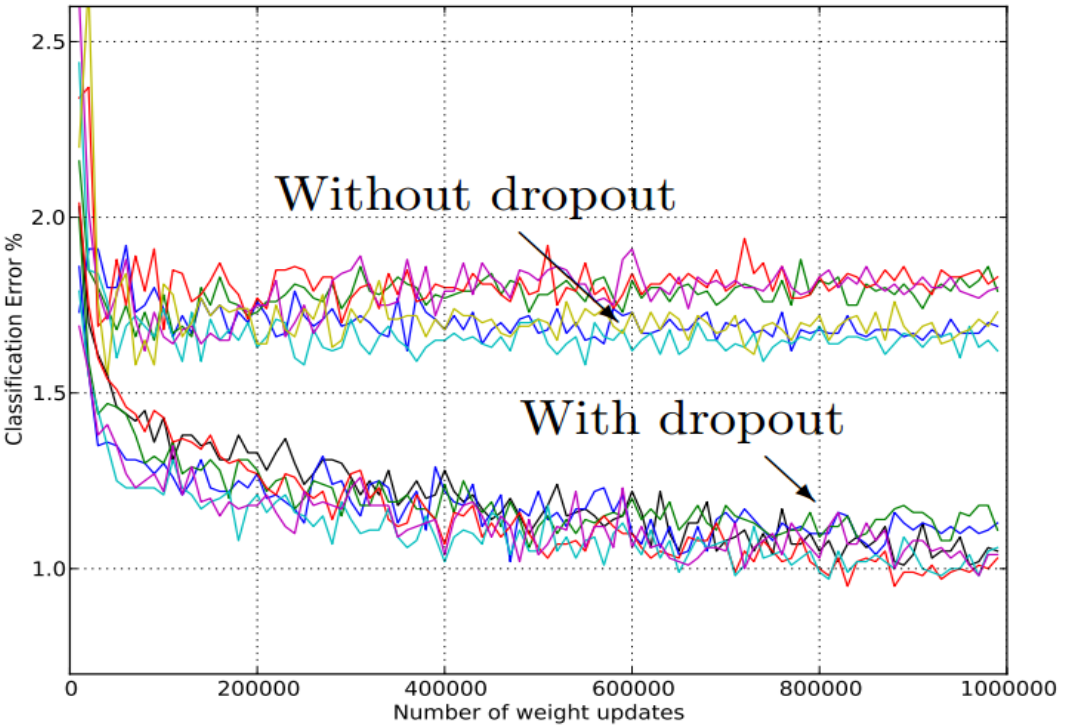
박성현

Index

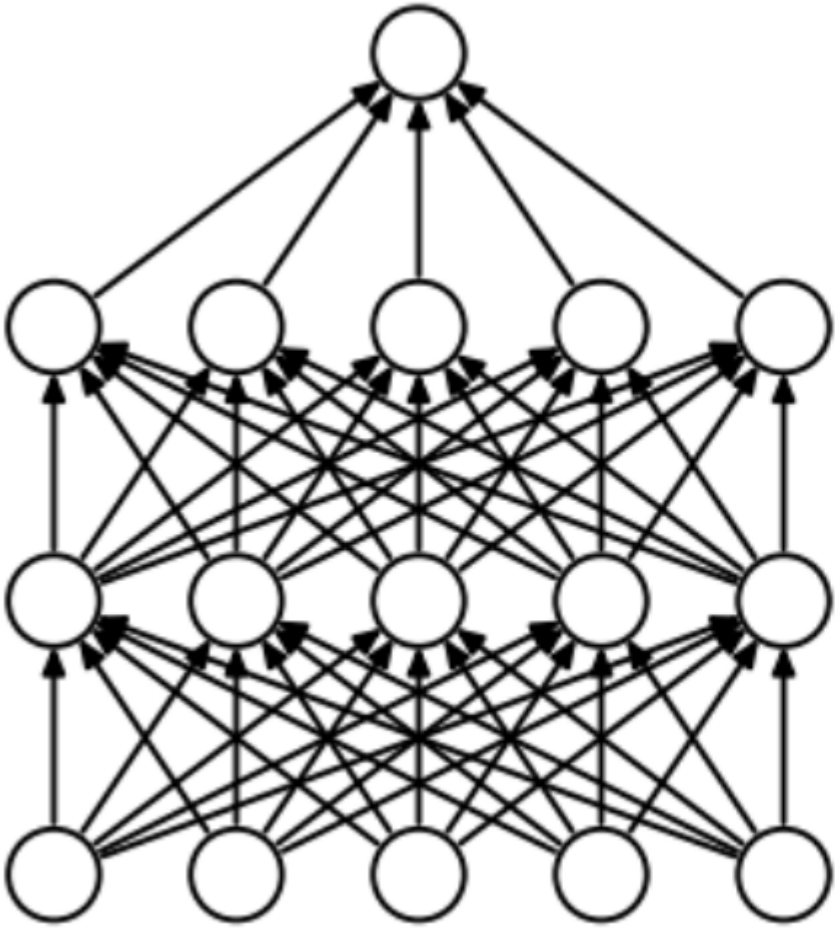
1. dropout / ensemble
2. RNN
3. LSTM

Dropout

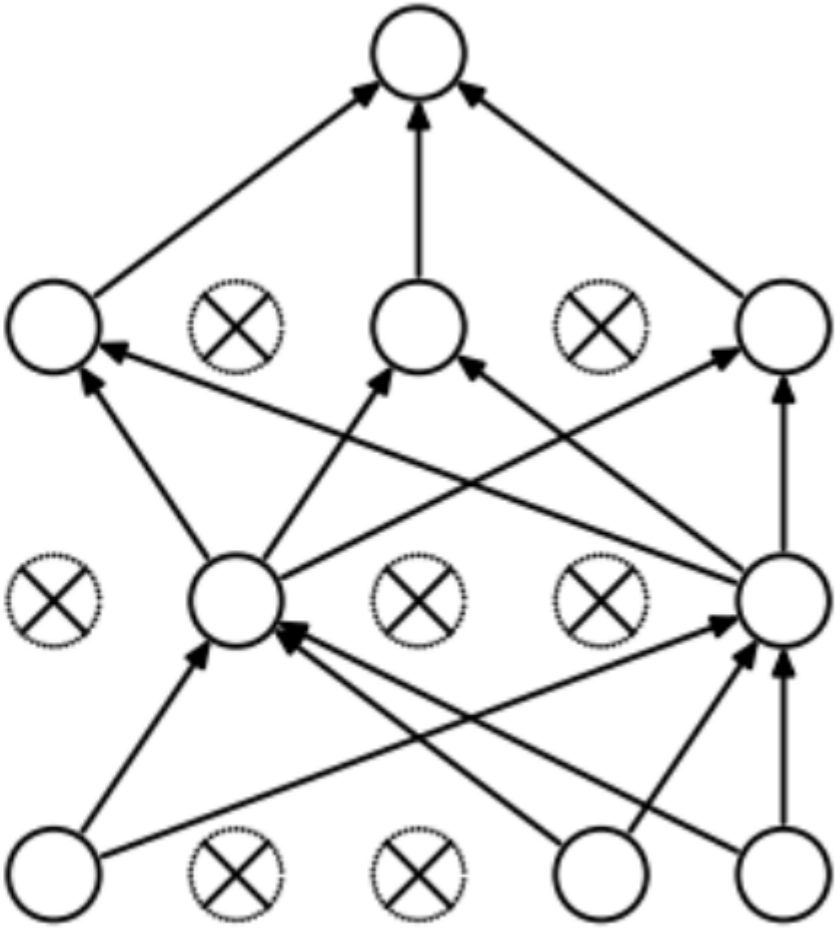
- Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks.
- A simple way to prevent overfitting



Dropout



(a) Standard Neural Net



(b) After applying dropout.

Dropout

TensorFlow implementation

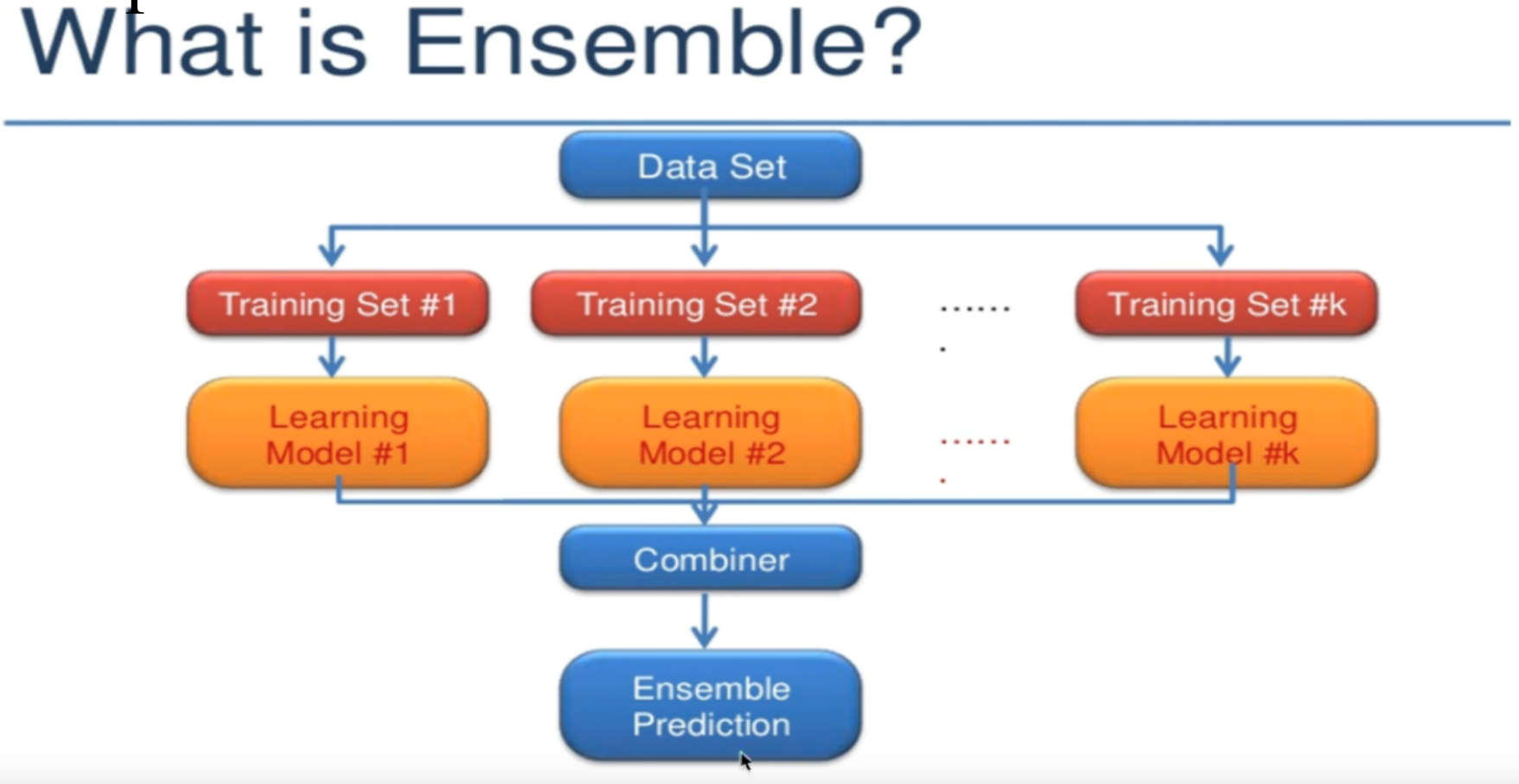
```
dropout_rate = tf.placeholder("float")
_L1 = tf.nn.relu(tf.add(tf.matmul(X, W1), B1))
L1 = tf.nn.dropout(_L1, dropout_rate)
```

```
TRAIN:
  sess.run(optimizer, feed_dict={X: batch_xs, Y: batch_ys,
  dropout_rate: 0.7})


EVALUATION:
  print "Accuracy:", accuracy.eval({X: mnist.test.images, Y:
  mnist.test.labels, dropout_rate: 1})
```

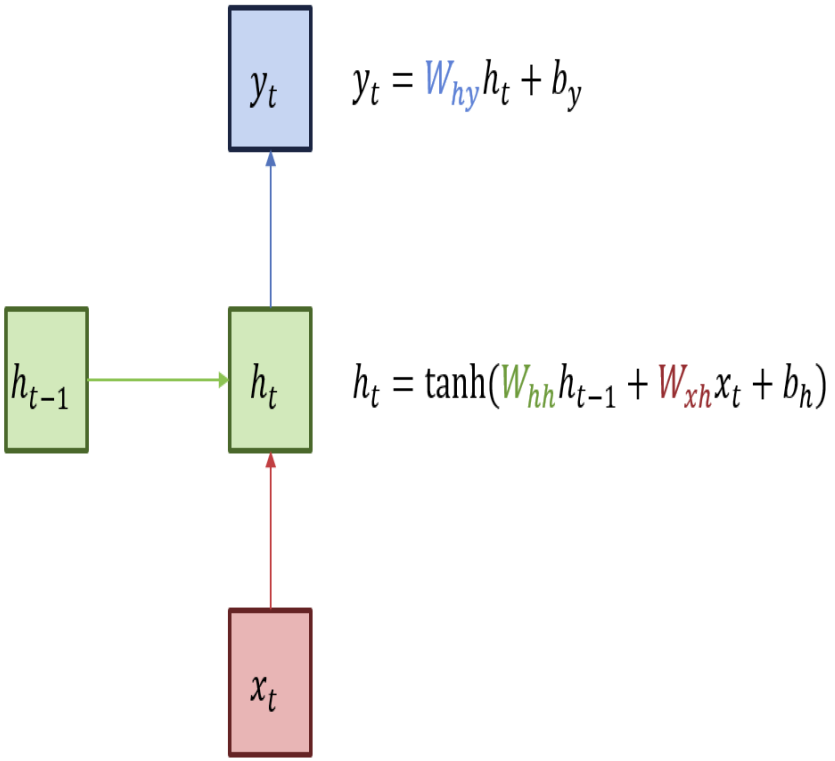
Ensemble

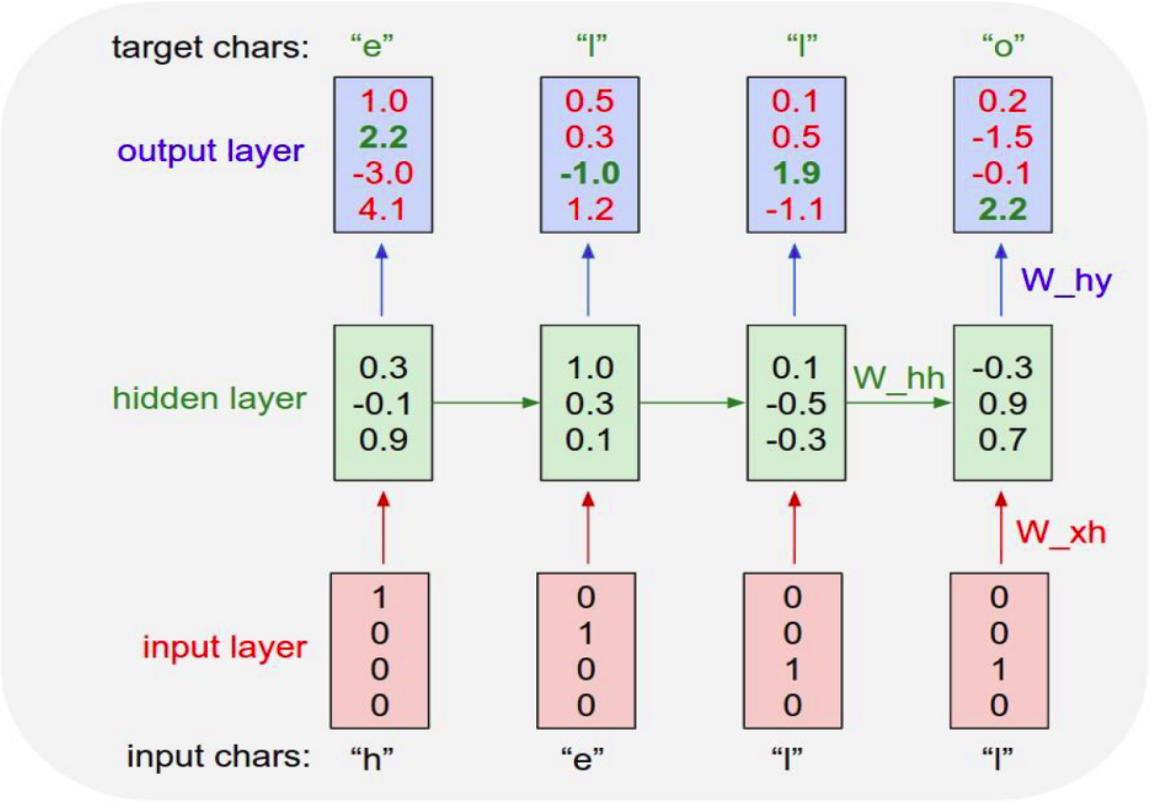
- Ensemble model definition is very straightforward. It uses the same input layer that is shared between all previous models.



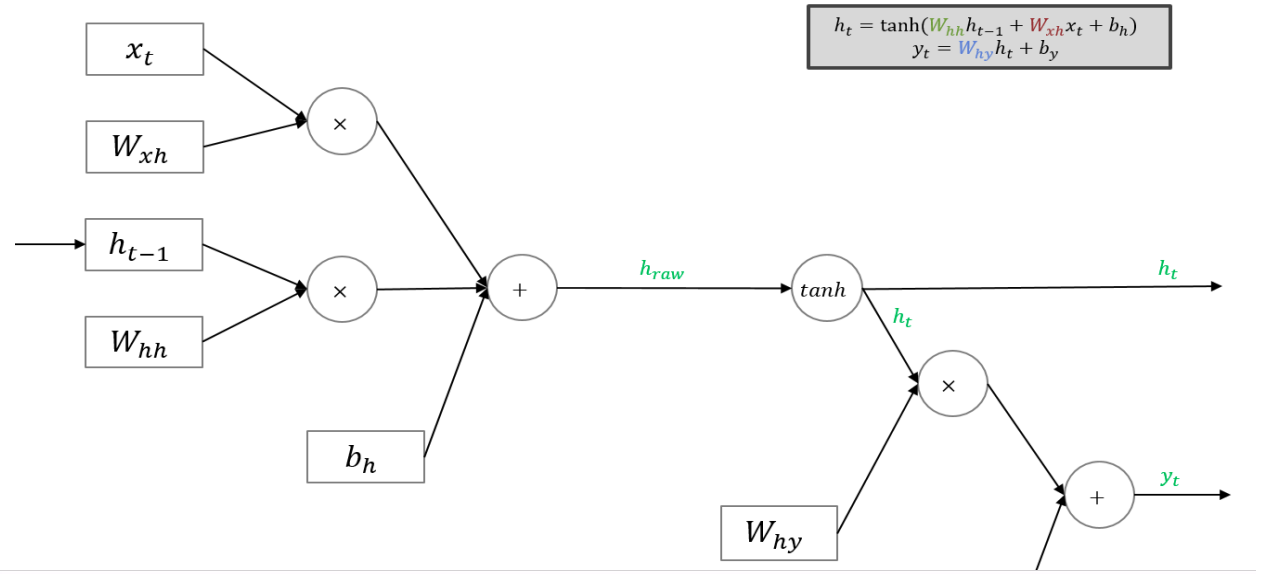
RNN

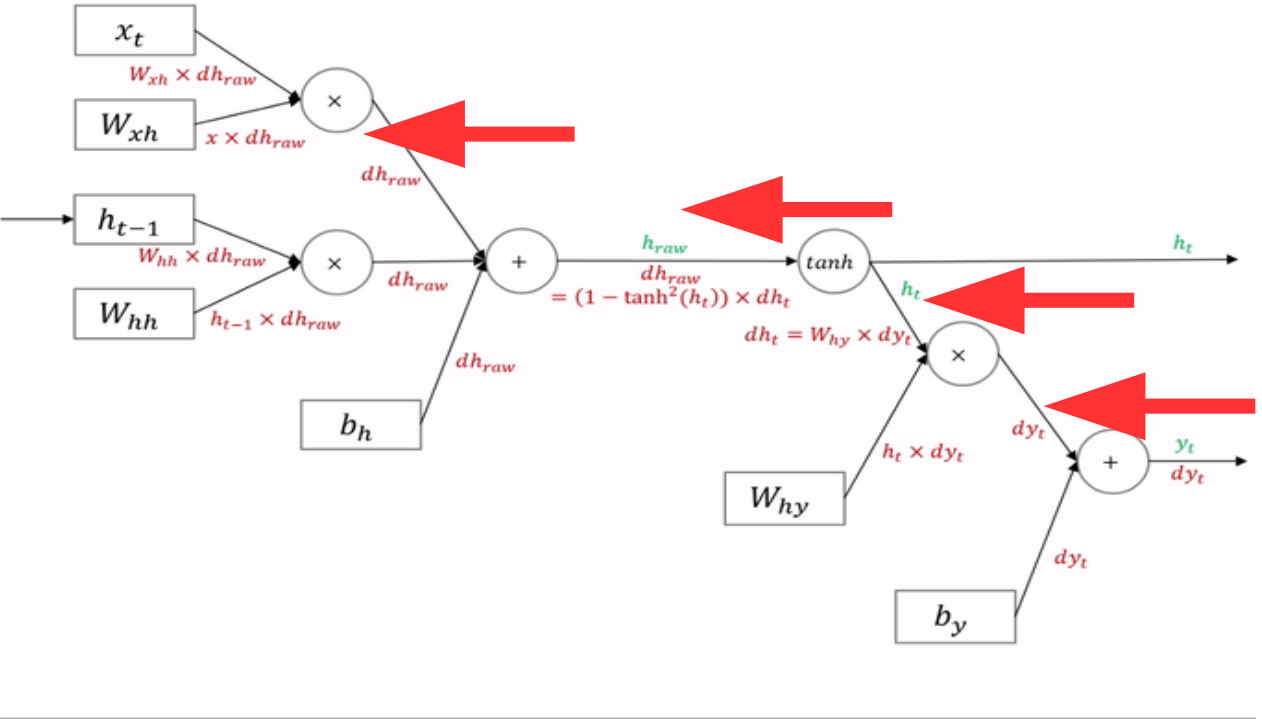
- What is RNN?





Forward / Back - Propagation



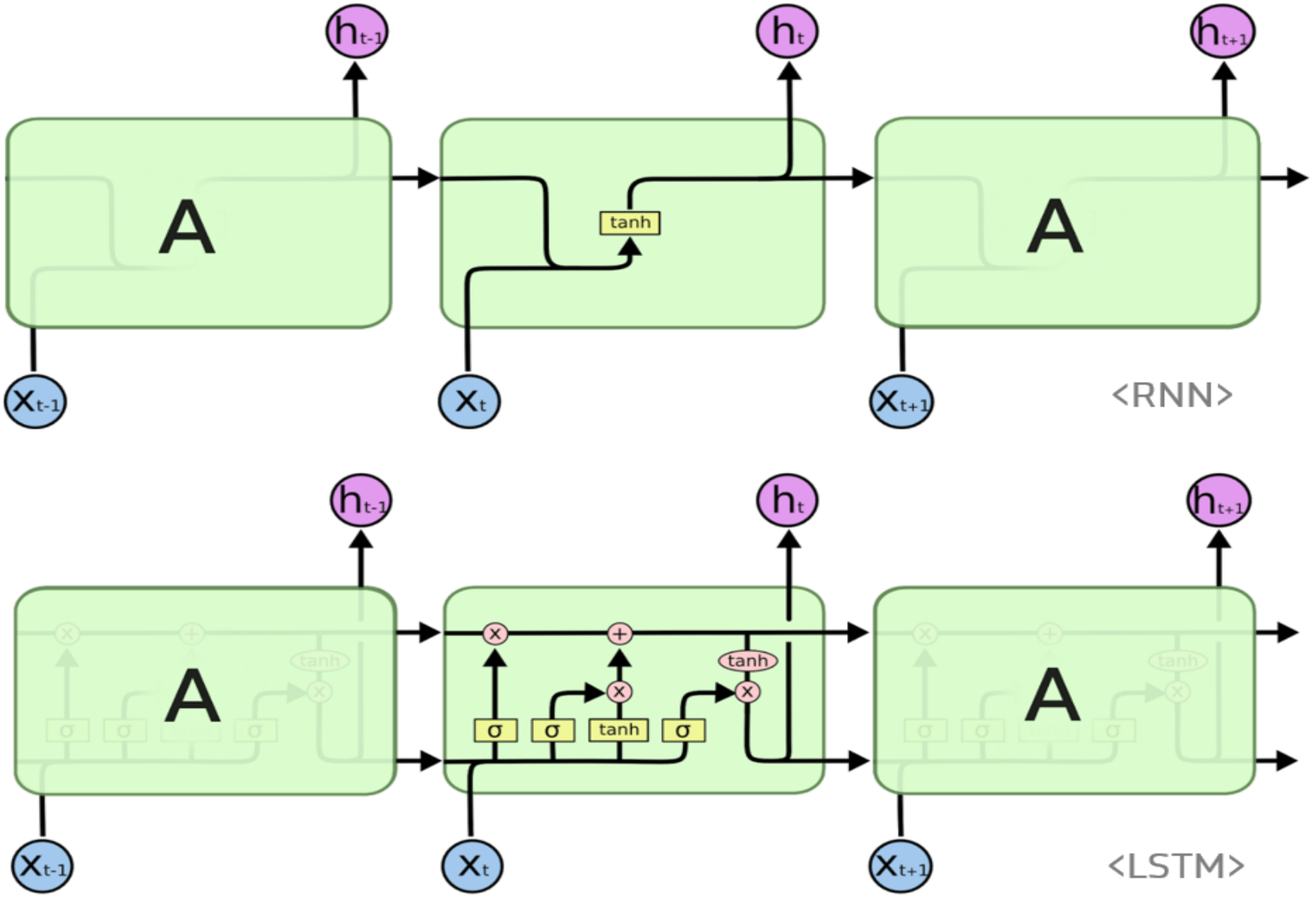


Useful Neural Network for sequential Data set
[ex. Sentences, Conversation, Stock...]

However...RNN is

- If the distance btw each step is long…
- Back-propagation gradient becomes 0 → useless
  ["Vanishing Gradient Problem"]
- To solve this problem… LSTM is widely used

LSTM


h_{t-1}
h_t
h_{t+1}
A
tanh
A
X_{t-1}
X_t
X_{t+1}
<RNN>
h_{t-1}
h_t
h_{t+1}
A
tanh
σ
σ
tanh
σ
tanh
A
X_{t-1}
X_t
X_{t+1}
<LSTM>

LSTM

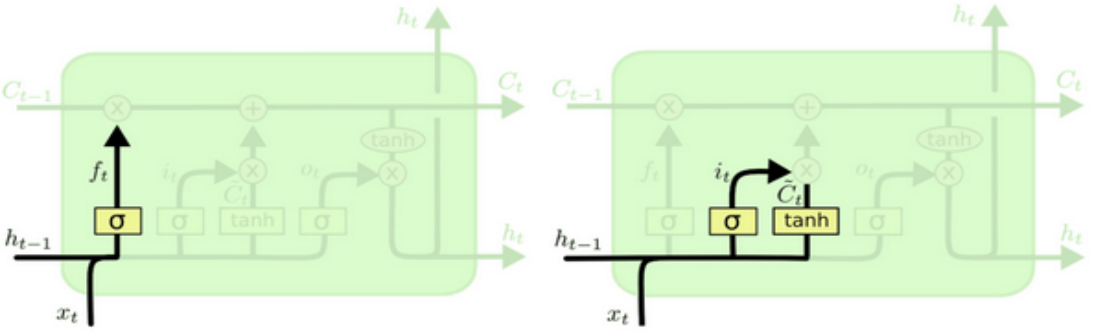
- How it works?

cell state는 일종의 컨베이어 벨트 역할을 합니다. 덕분에 state가 꽤 오래 경과하더라도 그래디언트가 비교적 전파가 잘 되게 됩니다. LSTM 셀의 수식은 아래와 같습니다. ⊙는 요소별 곱셈을 뜻하는 Hadamard product 연산자입니다.

$$
\begin{aligned}
f_t &= \sigma(W_{xh\_f}x_t + W_{hh\_f}h_{t-1} + b_{h\_f}) \\
i_t &= \sigma(W_{xh\_i}x_t + W_{hh\_i}h_{t-1} + b_{h\_i}) \\
o_t &= \sigma(W_{xh\_o}x_t + W_{hh\_o}h_{t-1} + b_{h\_o}) \\
g_t &= \tanh(W_{xh\_g}x_t + W_{hh\_g}h_{t-1} + b_{h\_g}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

forget gate f_t 는 '과거 정보를 잊기'를 위한 게이트입니다. h_{t-1} 과 x_t 를 받아 시그모이드를 취해준 값이 바로 forget gate가 내보내는 값이 됩니다. 시그모이드 함수의 출력 범위는 0에서 1 사이이기 때문에 그 값이 0이라면 이전 상태의 정보는 잊고, 1이라면 이전 상태의 정보를 온전히 기억하게 됩니다.

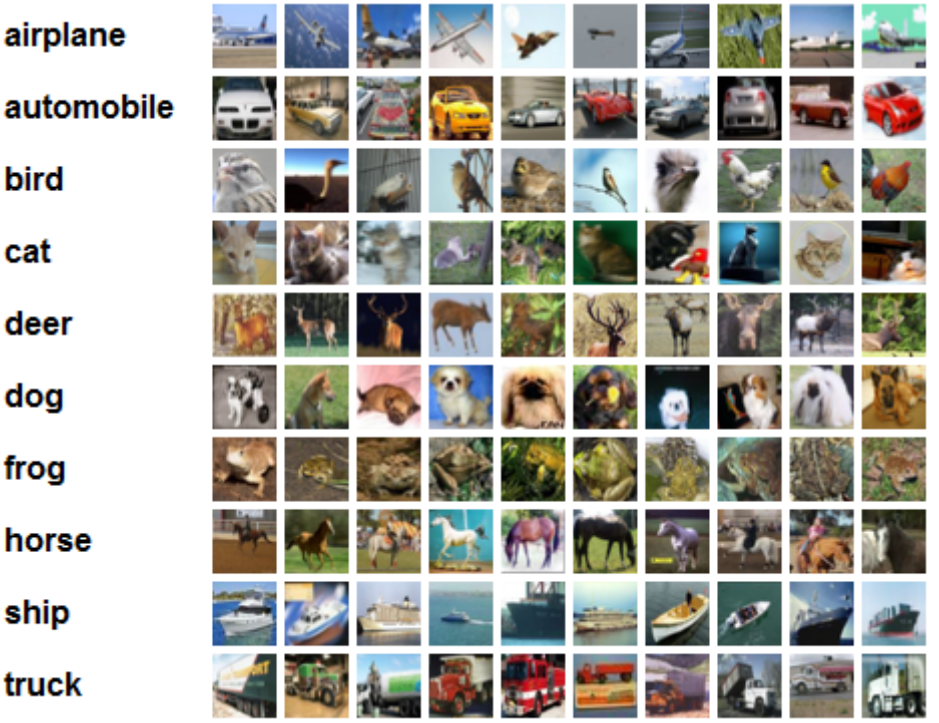
input gate $i_t \odot g_t$ 는 '현재 정보를 기억하기' 위한 게이트입니다. h_{t-1} 과 x_t 를 받아 시그모이드를 취하고, 또 같은 입력으로 하이퍼볼릭탄젠트를 취해준 다음 Hadamard product 연산을 한 값이 바로 input gate가 내보내는 값이 됩니다. 개인적으로 i_t 의 범위는 0~1, g_t 의 범위는 -1~1이기 때문에 각각 강도와 방향을 나타낸다고 이해했습니다.



<forget gate> <input gate>

HW - CIFAR!

- What is CIFAR?
- Canadian Institute For Advanced Research
- CIFAR-10 & CIFAR-100 [10 labels or 100 labels]



HW - CIFAR!

- 1] do implementation of CNN to classify CIFAR-10
- 2] Goal : 90% of Accuracy