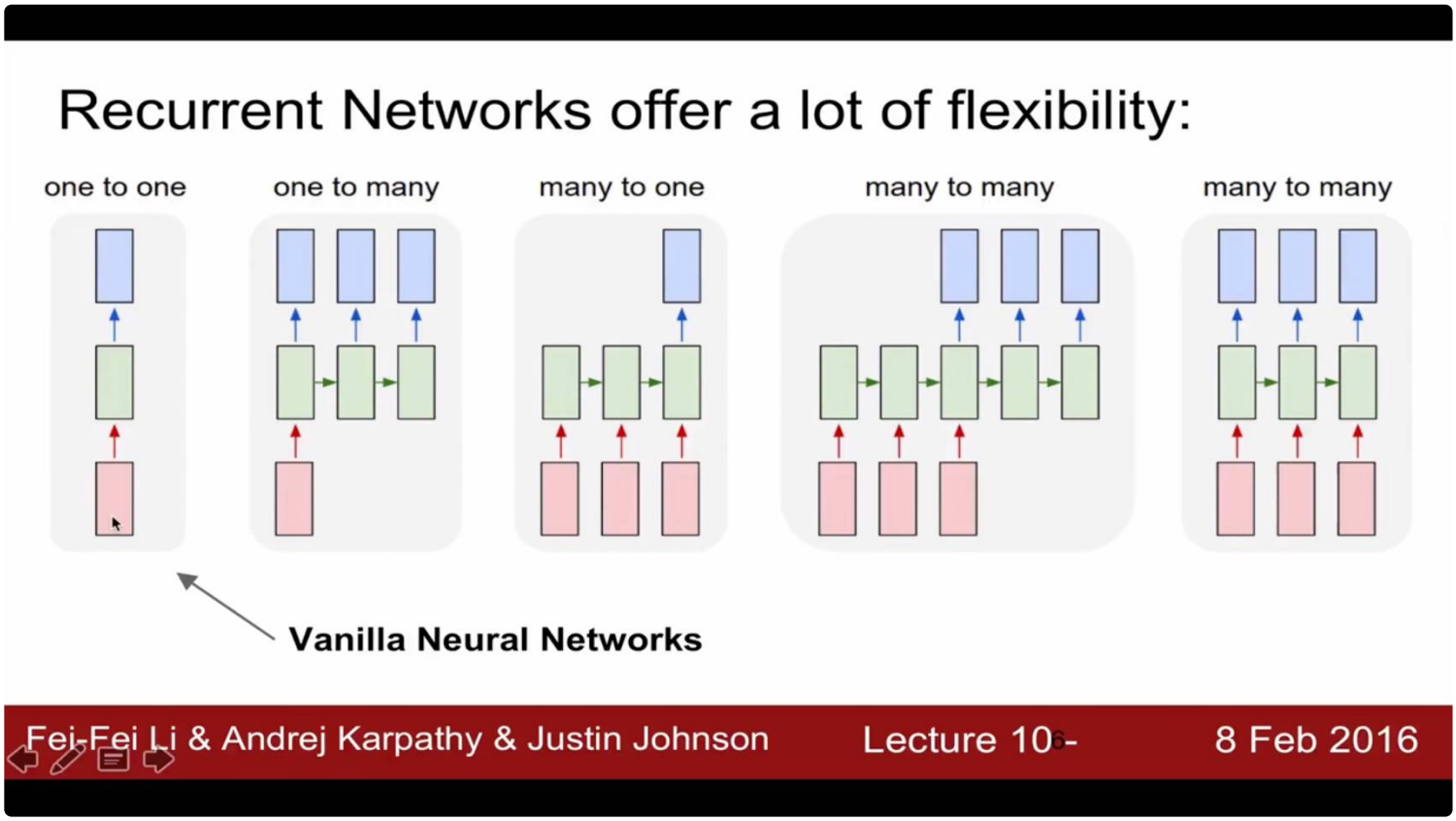
image optimazation 넘김

Recurrent network



- vanilla neural networks

input layer hidden layer output 으로 구성됨

input layer 에는 fixed 된 사이즈의 이미지가 들어갈 것이고

output layer 에서도 fixed 된 사이즈의 output 벡터가 나오게 될것

- rnn 의 경우에는 이런식으로 sequence 가 존재한다.
- one to many 는 output 에 sequence
- many to one 은 input 에 sequence
- many to many 는 input 과 output 에 sequence

rnn 의 각각의 예에 대해서 알아볼 것이다.

one to many 의 경우 대표적인 예에는 image captioning 이 있다.

이미지를 설명하는 단어들의 시퀀스를 말한다.

many to one 의 경우에는 sentiment classification 이 있습니다. 감정을 분류해 내는 것인다. 감정으로 분류되는 단어들이 input 에 주어졌을 때 이것이 positive 하냐 negative 하냐를 output 으로 내놓습니다.

many to many 의 경우 translation 이 된다. 영어 단어로 구성 된 문장이 들어왔을 때 한국어로 뱉어주는 방법에 대한 것이 된다.

그리고 마지막 그림의 many to many 의 경우 비디오 클래시피케이션이 해당한다. 비디오의 경우에는 프레임마다

예측이 이루어 질 것이다.

이 함수를 만들때 유의해야 하는 것이 있는데 예측이 지금 현재의 프레임에만 국한되어서는 안된다는 것이다.

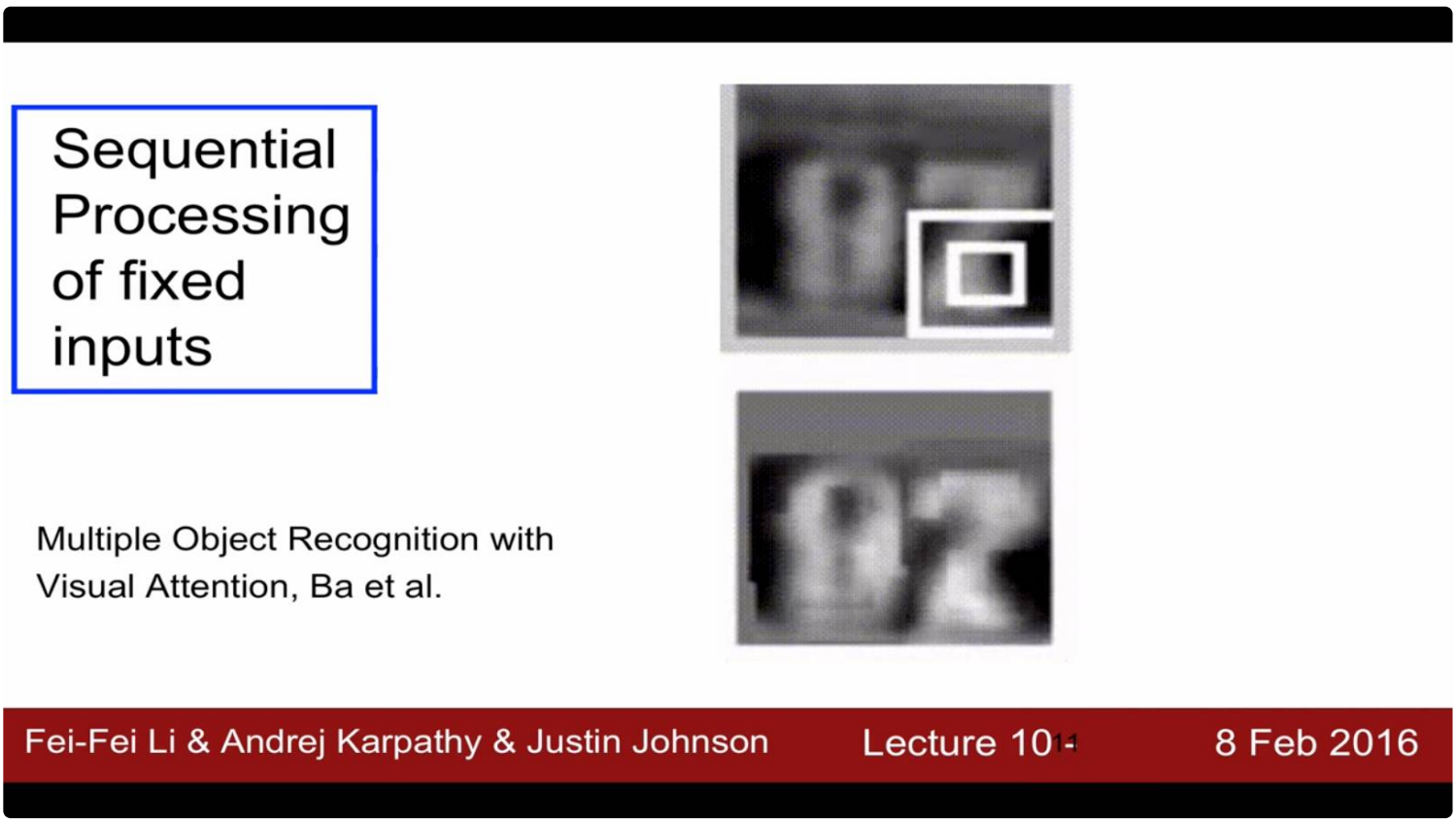
비디오에 있어서 예측은 현재의 프레임 뿐만아니라 이전의 프레임 전부를 위한 함수가 되어야 한다는 것이다.

모든 각각의 타임 스탭에 대한 예측은 현재의 프레임 + 지난 프레임들의 함수로 이루어져야 한다.

이런 원투원인 경우에는 인풋이나 아웃풋이 시퀀스가 아닌 형태지만

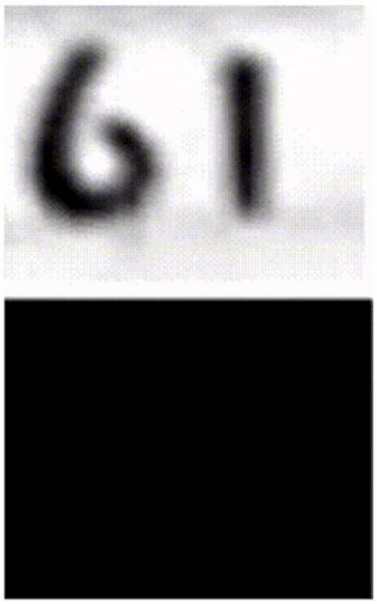
픽스드 된 형태를 가지므로 이 인풋 아웃풋에 대해 각각이 시퀀스하다고 처리하고 시퀀셜한 처리를 해줄 수도 있다는 것을 이해하자

fixed inputs 를 시퀀셜하게 만들어 낸것



Sequential Processing of fixed inputs

Multiple Object Recognition with Visual Attention, Ba et al.

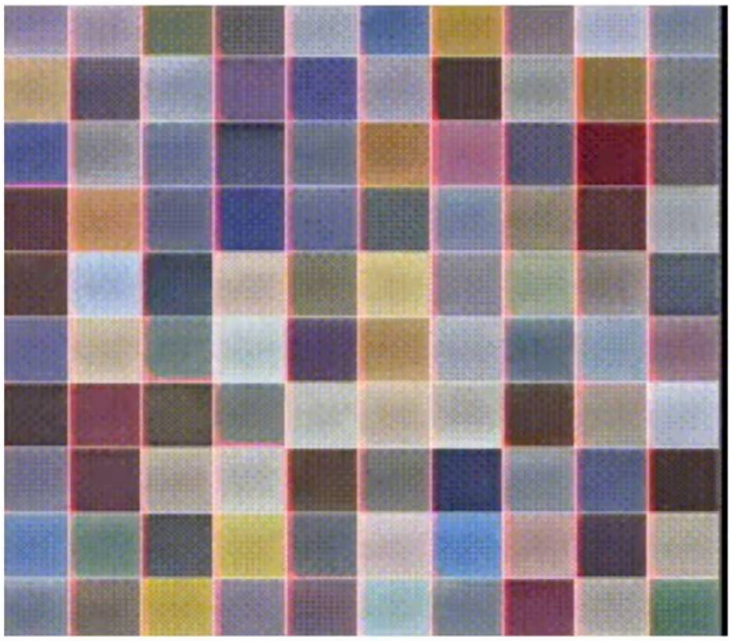


영상을 보면 그림을 그리듯이 위 아래로 진동하면서 글씨를 만들어가는 모습을 확인할 수 있는데

이는 시퀀셜하게 이미지를 훑어나가는 과정을 보여주는 것이다.

Sequential Processing of fixed outputs

DRAW: A Recurrent Neural Network For Image Generation, Gregor et al.



이는 fixed 사이즈의 output 을 시퀀셜하게 나타낸 것이다.

Sequential Processing of fixed outputs

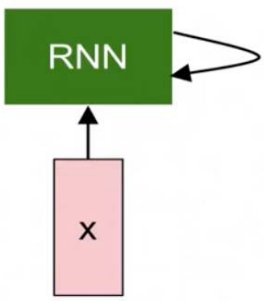
DRAW: A Recurrent Neural Network For Image Generation, Gregor et al.

숫자가 한번에 나타나는 것이 아니라 한쪽에서 써 내려나가는 것 처럼

out put 이 출력되게 된다.

recurrent 뉴럴 넷을 이렇게 사용할 수 있다는 것

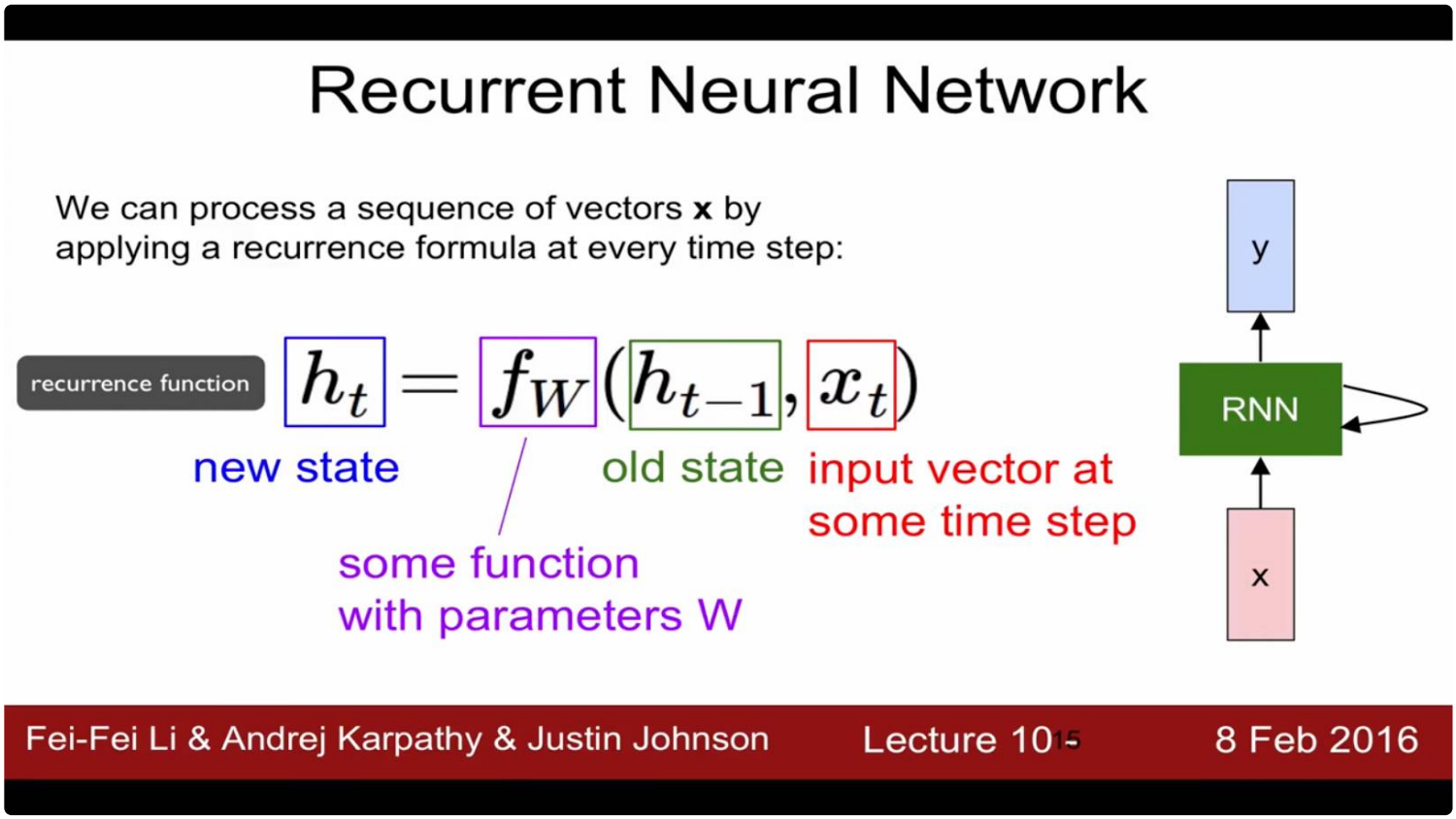
Recurrent Neural Network



x 는 인풋 벡터이다.

매번 인풋 벡터가 이렇게 입력되는데 이 상태를 인풋 벡터를 받는 function 으로 만들어 줄 수 잇는데. 이게 weight 로 구성되고 새로운 인풋이 들어올 때 마다 다른 반응을 받을 수 있다.

매타임스탬마다 입력되는 어떤 벡터 x 에서



이와 같은 리쿼런스 함수를 적용할 수 있다.

state update 는 이런식으로 이루어 진다.

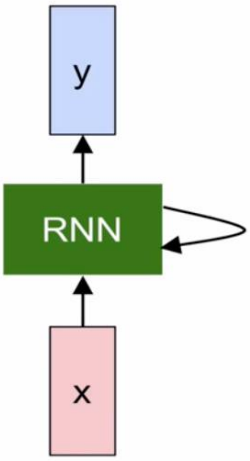
직전의 히든 스테이트 + 지금 타임 스탭

w 값들을 학습 시켜나가는 것

매 타임 스탭마다 동일한 펑션, 동일한 파라미터 셋들이 사용되어야 한다. 인풋의 시퀀스 사이즈 아웃풋의 시퀀스 사이즈에 구애받지 않고 사용 가능하다.

(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

상태가 단일의 hidden vector h 로 구성되어있음

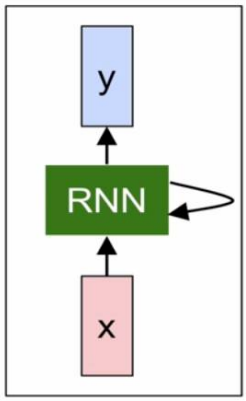
whh 직전의 히든 레이어 현재의 히든 레이어

wxh 어떤 새로운 입력값에 대해 값이 바뀐다 라는 것

Character-level language model example

Vocabulary:
[h,e,l,o]

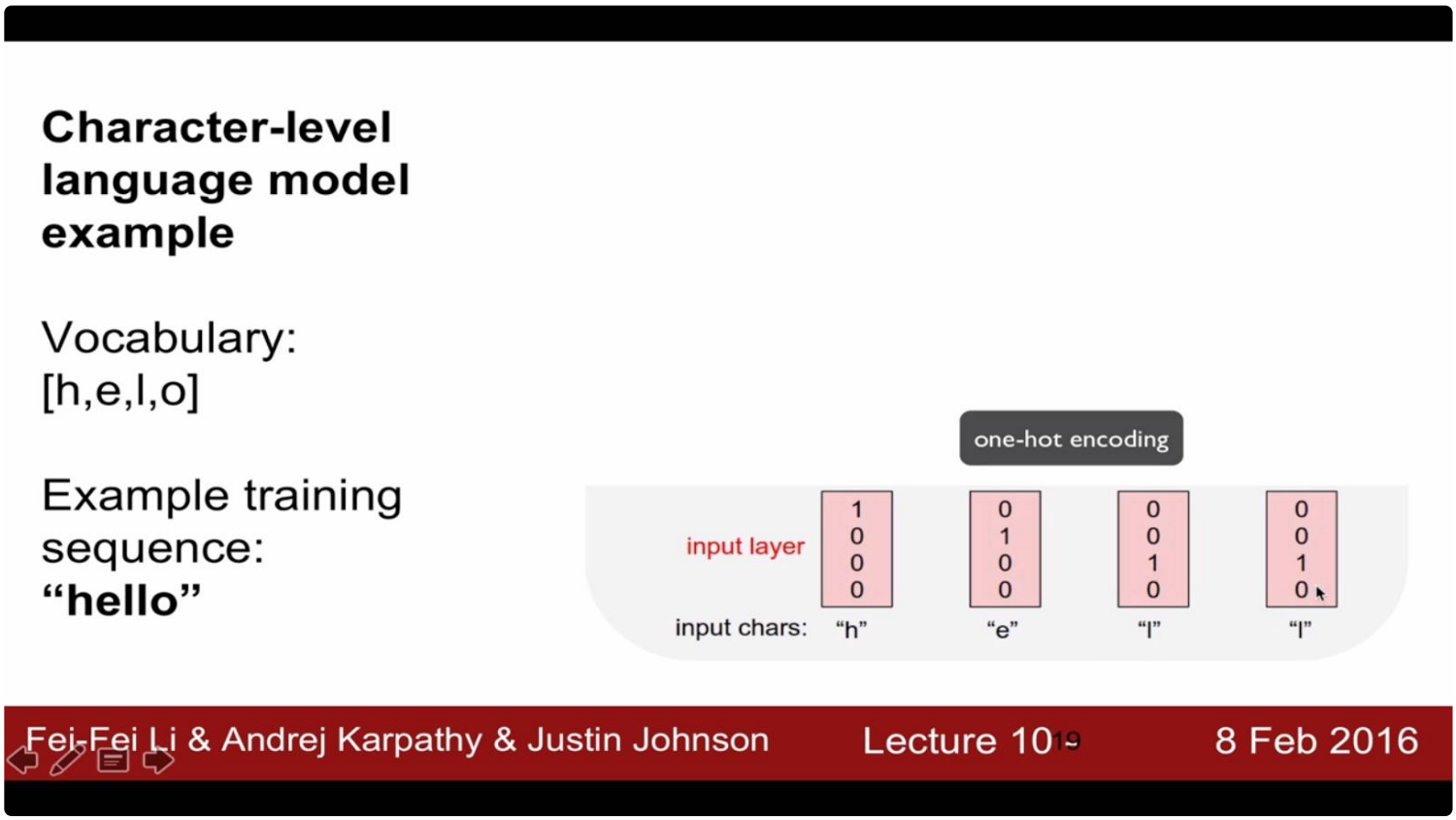
Example training sequence:
"hello"



우리가 학습 시키고자 하는 것은 이 hello 라는 단어이고

매 타임스탭마다 물어보게 됩니다. 다음에 어떤 단어가 올 것 같냐고

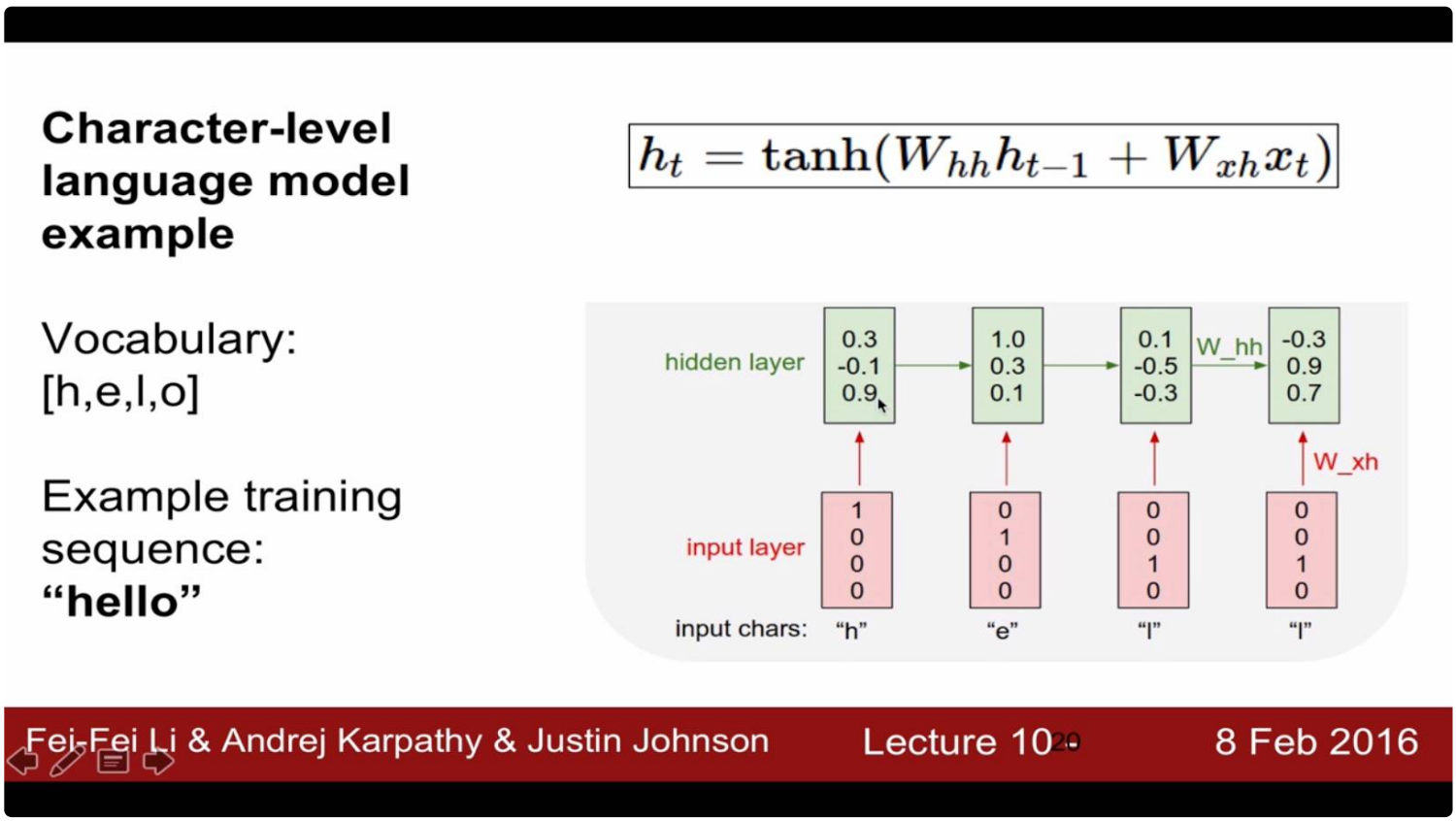
이렇게 물어보면서 전체적으로 진행되는 것이다.



원핫 인코딩을 통해 이런 데이터를 입력해 주었다고 가정하자

input 레이어는 이렇게 구성될 것이고

hidden layer 는



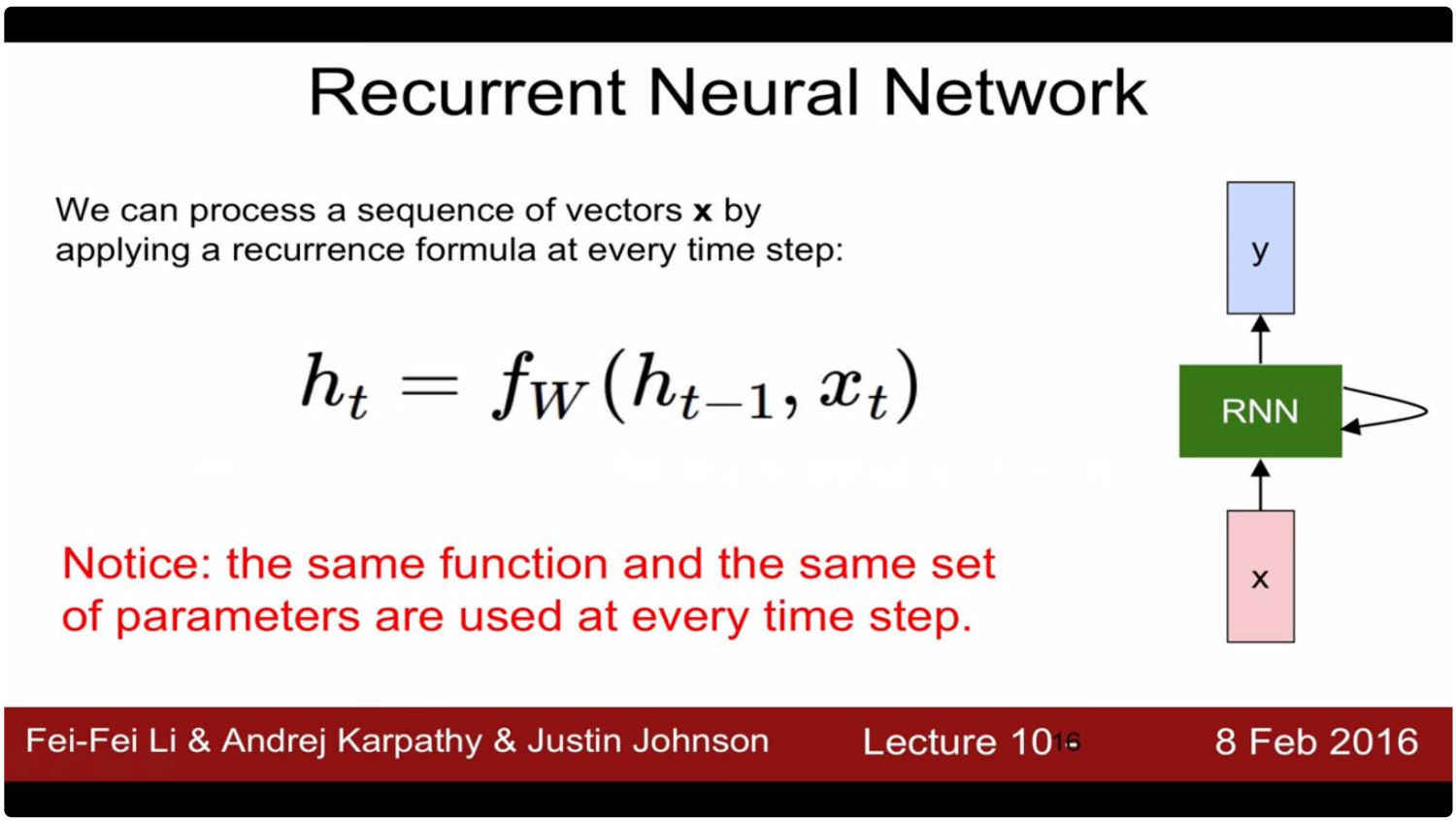
3개의 뉴런들로 구성되는 히든 레이어라고 가정을 할건데

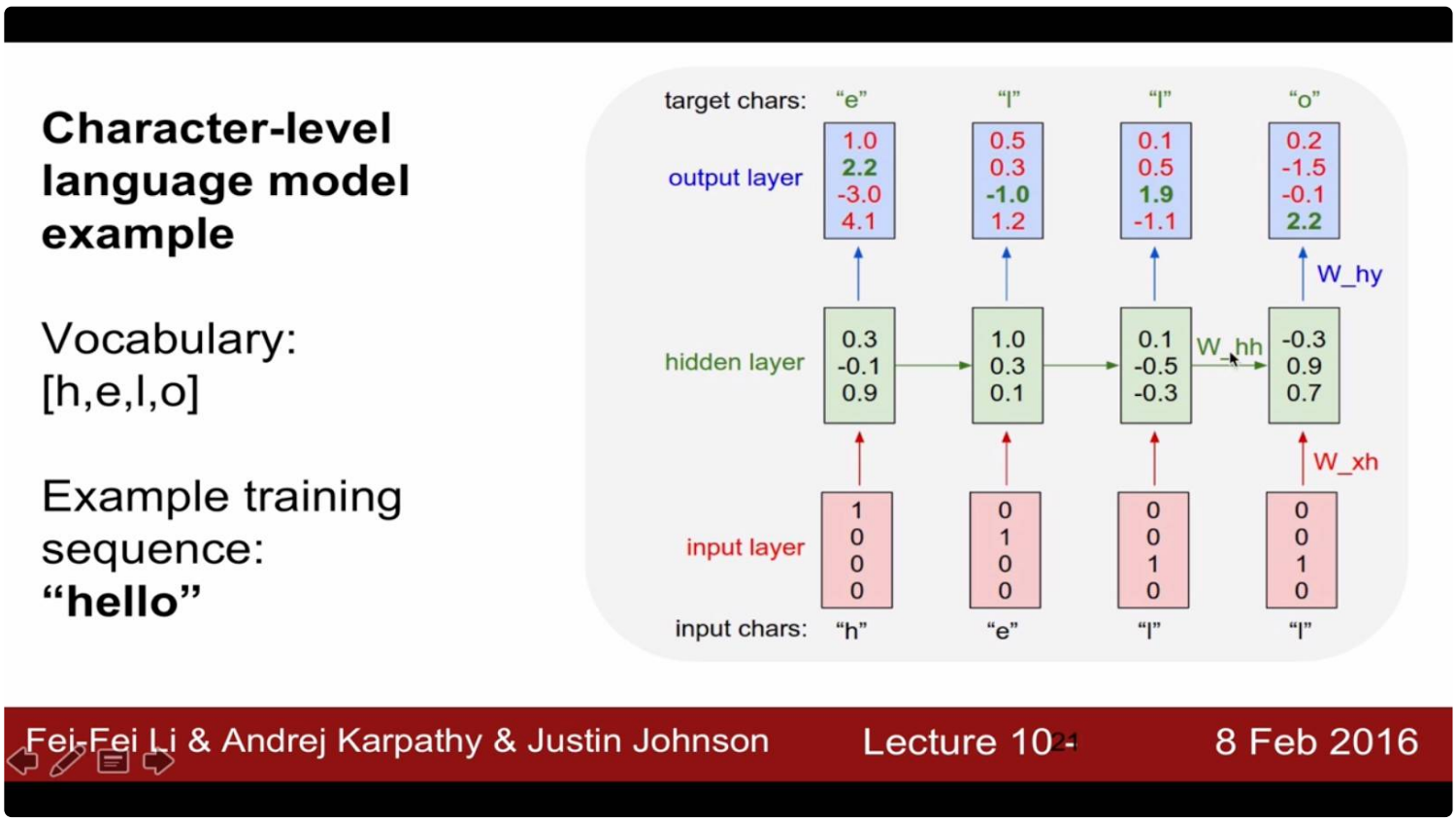
이 히든 레이어는 이전의 타임 스탭의 영향을 받아 반영되게 된다.

이 초록색 방향으로 진행하는 것이 W_hh 히든 레이어에서 히든 레이어로 가는 것이고

빨간색은 x 에서 hidden layer 로 파란색은 히든 레이어에서 y 로 가는 것이다.

모든 타임 스탭에 대해서 우리는 모두 동일한 파라미터를 사용한다는 말을 하였다.





따라서 위 그림의 이 파란 선 빨간선 초록 선 들은 각각 서로 다 동일한 것이다.

그렇게 때문에 인풋의 시퀀스 수와 아웃풋의 시퀀스의 수에 상관없이 처리가 가능하다는 것이다.

타겟 charecter 와 현재 output 레이어를 비교해보자

첫번째에서 제일 높은 값은 4.1 이므로 o 에 해당한다.

하지만 나온 결과 목표값인 e 와 다르므로 틀린 것이다.

두번째를 보자 1.2 가 젤 높으므로 o 로 예측된것이고 o 로 예측한것이니 목표값인 l 과 다르므로 틀렸다.

좀 더 이해를 높이기 위해 코드를 살펴보자

char 레벨의 rnn 은 이렇게 비교적 짧은 코드로(다른 것들에 비하면) numpy 를 사용하여 구현이 가능하다.

min-char-rnn.py gist

Data I/O

```
"""
Minimal character-level Vanilla RNN model. Written by Andrej Karpathy (@karpathy)
BSD License
"""
import numpy as np

# data I/O
data = open('input.txt', 'r').read() # should be simple plain text file
chars = list(set(data))
data_size, vocab_size = len(data), len(chars)
print 'data has %d characters, %d unique.' % (data_size, vocab_size)
char_to_ix = { ch:i for i,ch in enumerate(chars) }
ix_to_char = { i:ch for i,ch in enumerate(chars) }
```

자세히 들여다보면 넘파이를 임포트 해주었고

파일을 읽어온다음

set 을 통하여 unique 한 데이터 리스트로 만들어준다.

중요한건 캐릭터와 인덱스를 변환할 수 있도록 하는 장치를 만들어 둔 것이다.

이런식으로 연관을 시켜둠

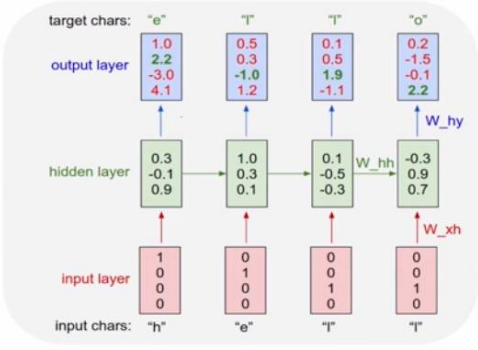
min-char-rnn.py gist

Initializations

```
# hyperparameters
hidden_size = 100 # size of hidden layer of neurons
seq_length = 25 # number of steps to unroll the RNN for
learning_rate = 1e-1

# model parameters
Wxh = np.random.randn(hidden_size, vocab_size)*0.01 # input to hidden
Whh = np.random.randn(hidden_size, hidden_size)*0.01 # hidden to hidden
Why = np.random.randn(vocab_size, hidden_size)*0.01 # hidden to output
bh = np.zeros((hidden_size, 1)) # hidden bias
by = np.zeros((vocab_size, 1)) # output bias
```

recall:

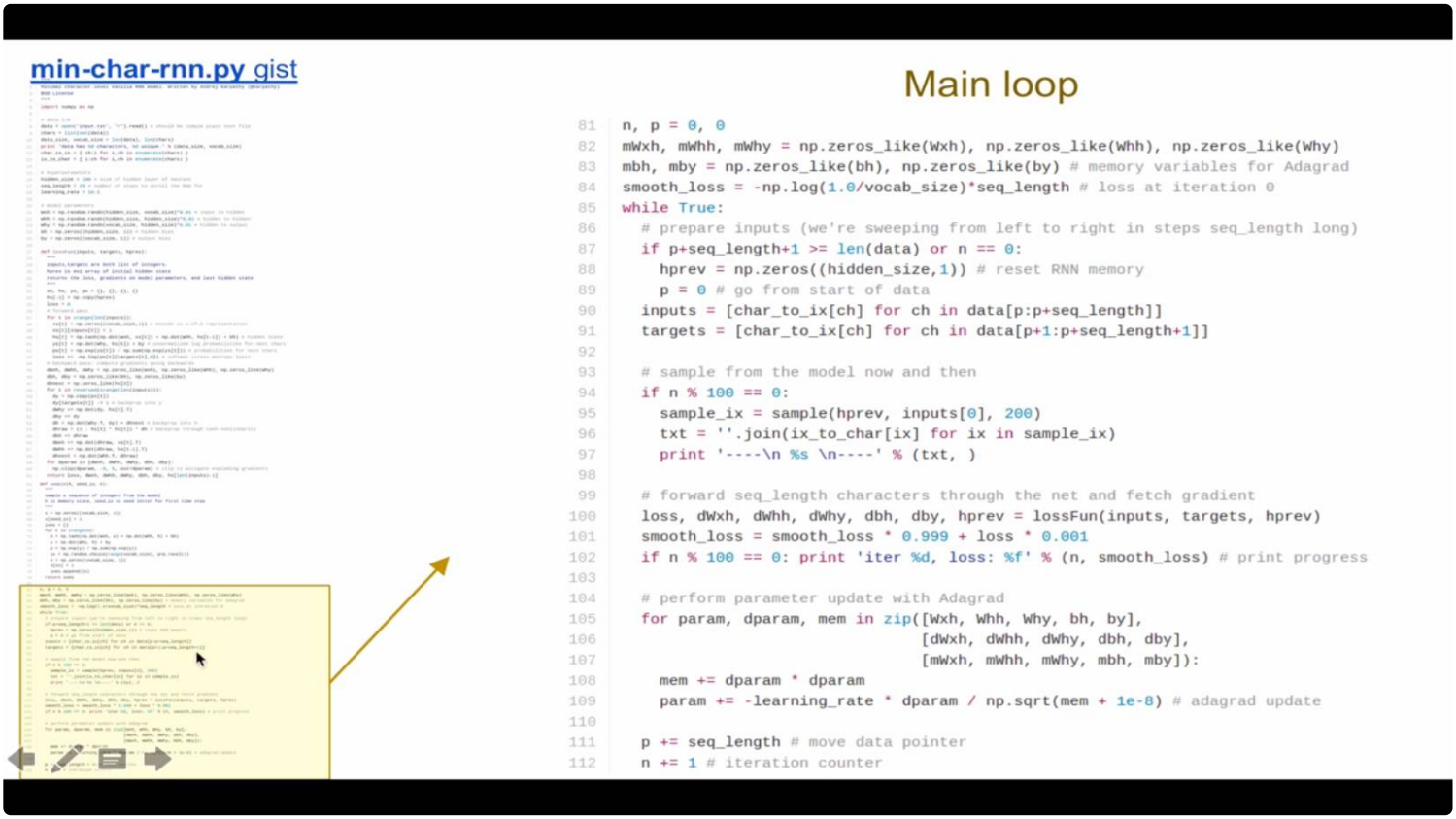


초기화 부분에서는 시퀀스의 길이가 너무 길다면 메모리에 한꺼번에 올리고 backpropagation 하는 것에 무리가 있기 때문에

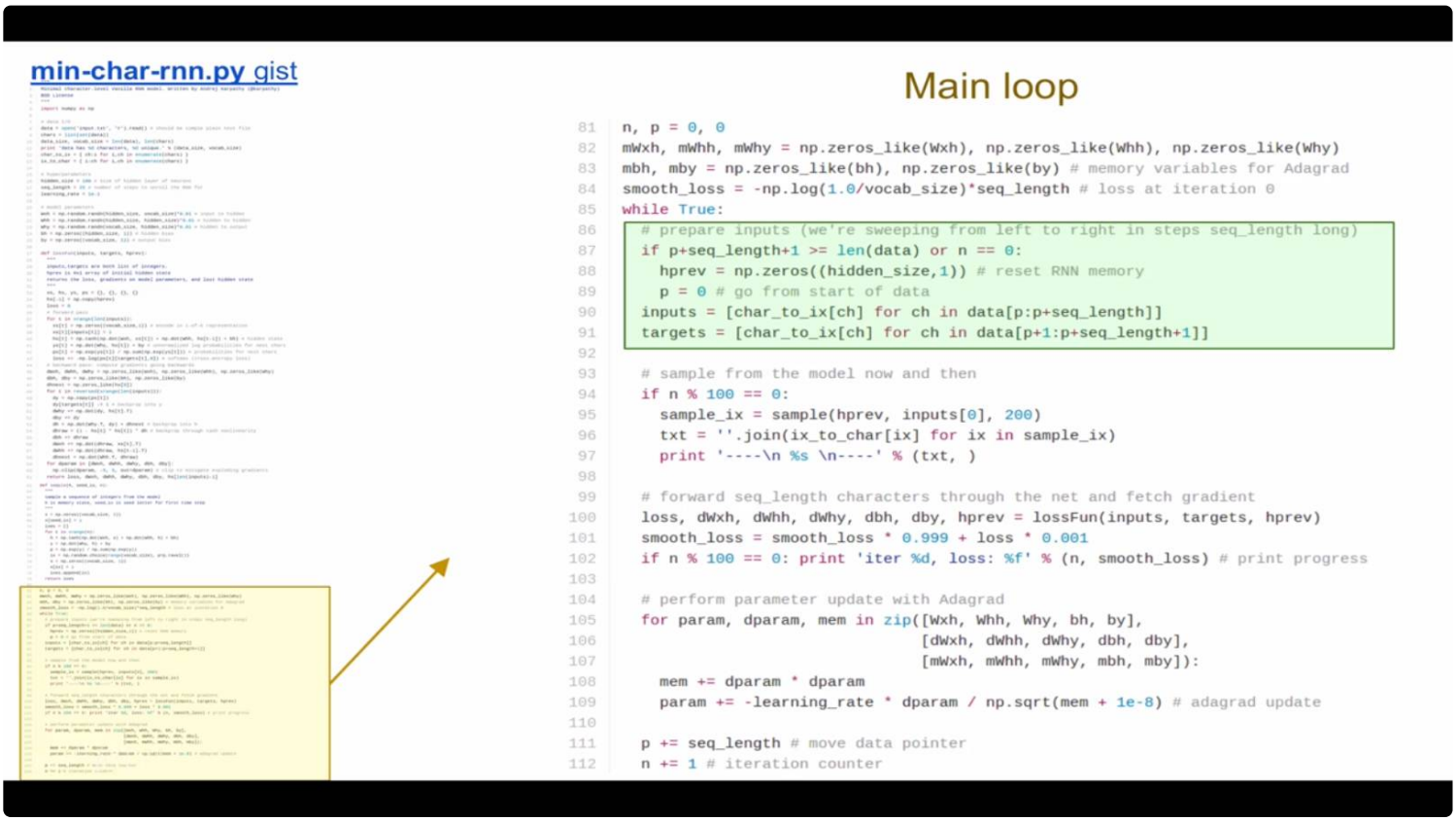
제약을 둔 것이다

이만큼 의 배치에 대해서 일단 처리를 해줘라 라고 제약둠 seq_length

hiddenlayer 의 값은 각각에 대해서 랜덤하게 추출해주고 있고



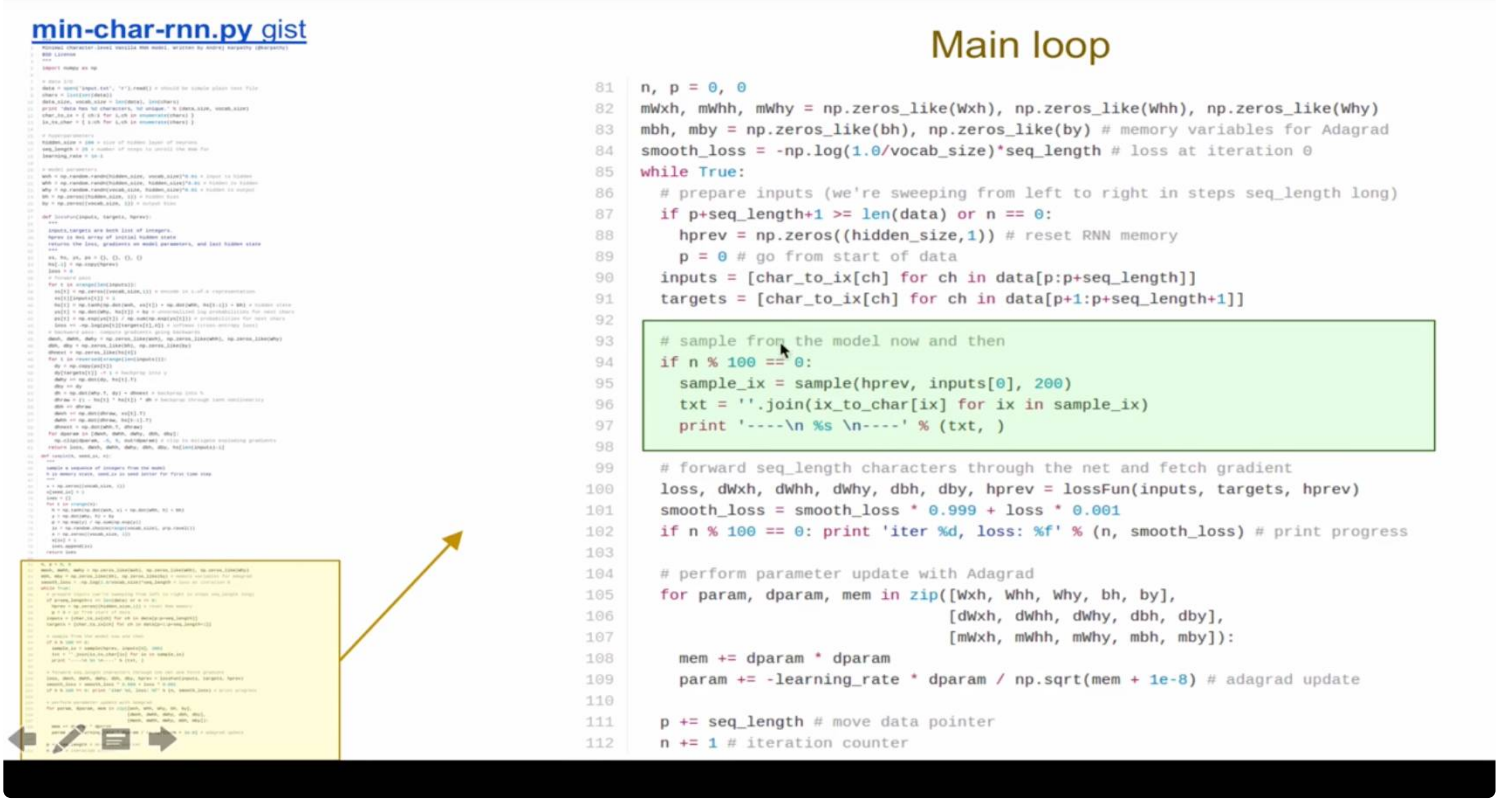
메인 루프의 상세한것에 대해서 하나 하나 살펴보겠습니다.



일단 rnn 메모리를 0으로 두고

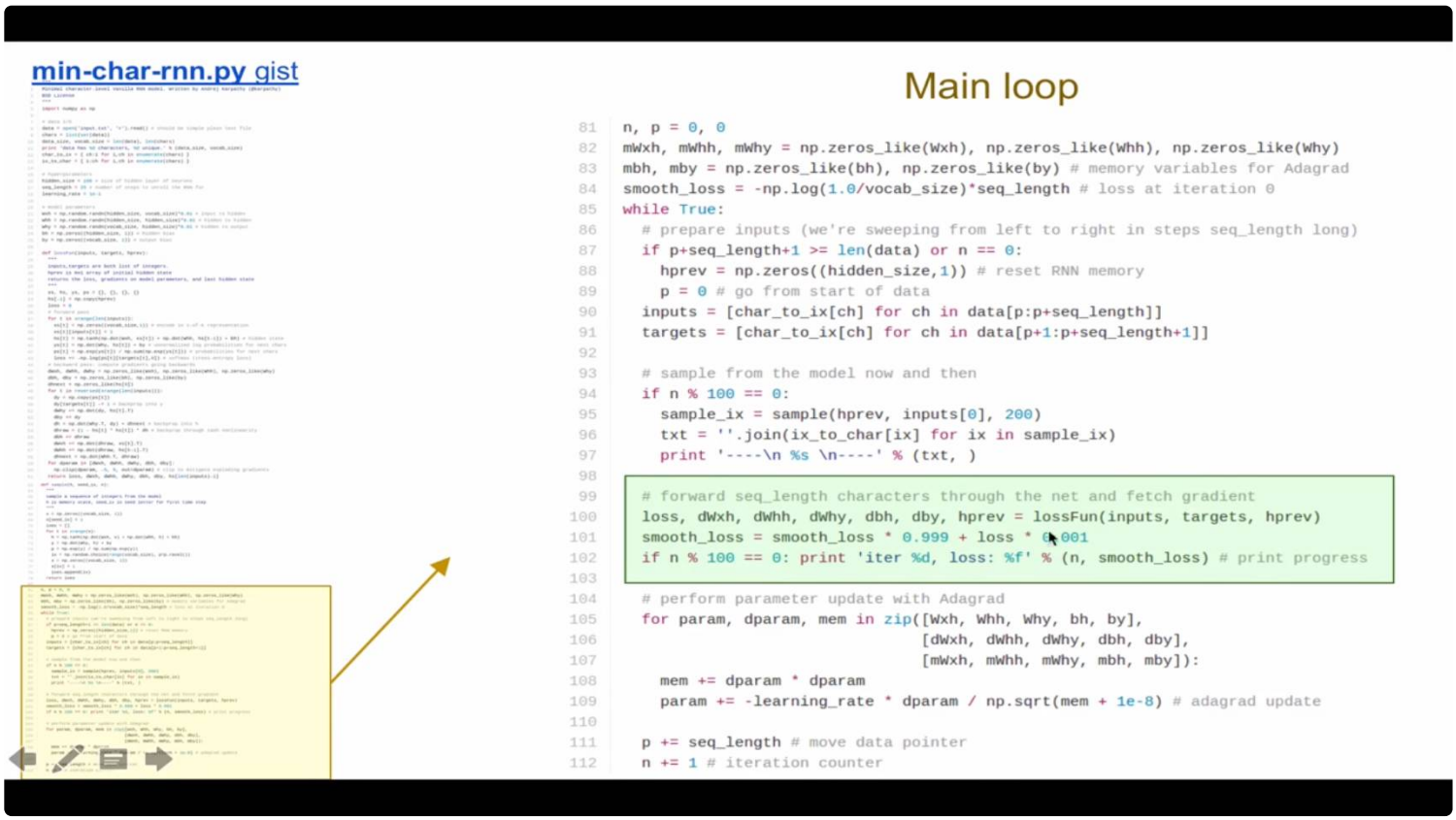
0 에서 길이까지 인풋

target 에서 +1 을 해주는것은 예측값을 넣어주기 위함이다.



다음 단계에서는 샘플링을 해준다.

샘플 함수를 호출하므로써 학습의 매 타임스탭마다 예측하는 캐릭터를 output 으로 출력하게 되는 것이다.

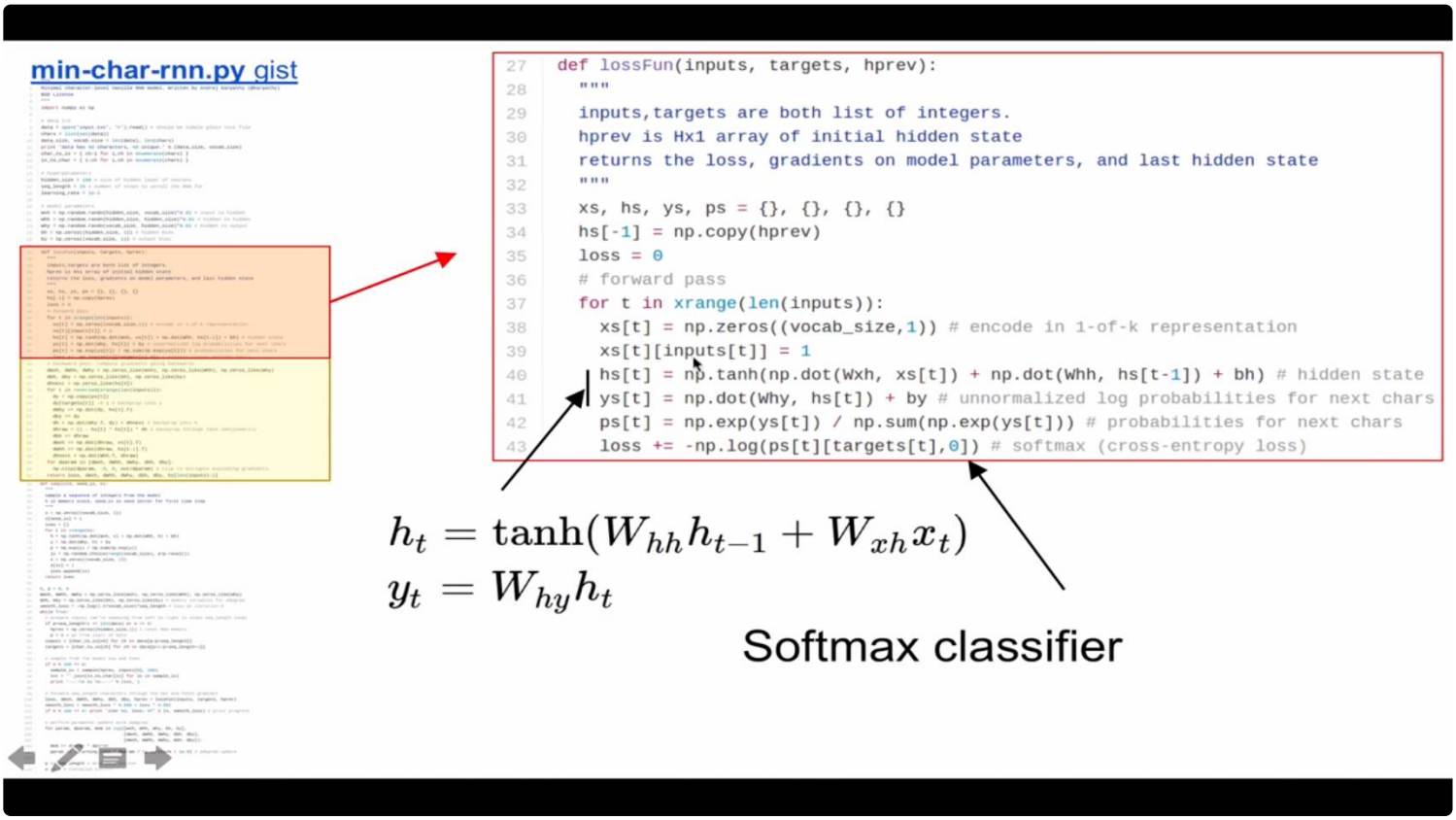


로스를 구할 때 우리가 넣어주는 것은 인풋 타겟 전단계의 값이다.

전단계의 히든 스테이트 벡터의 25개의 캐릭터 중에 가장 끝부분에 있는 것을 계속 추적하게 되는데 다음 배치가 feed 되었을 때 초기 백터를 계속 피딩해주는 것

그래서 다음 배치로 이어질때 잘 되도록 해준 것

그다음 파라미터 업데이트를 하면 포워드 패스, 백워드 패스가 존재 인풋값이 25



로스 펑션을 보면

25번 반복을 하게 되는데 인풋을 제외한 것은 모두 0으로 설정해준다.

hs 의 경우 나왔던 아래의 공식 처럼 연산을 수행해주고

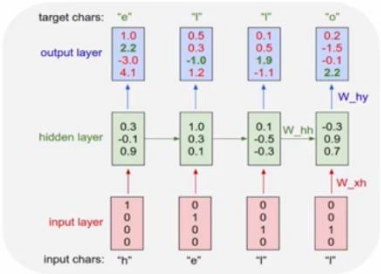
ys 도 마찬가지로 진행한다.

ps에 확률을 계산하여 대입하여 주고

softmax classifier 를 사용하여 로스를 계산해 준다.

min-char-rnn.py gist

```
# backward pass: compute gradients going backwards
dWxh, dWhh, dWhy = np.zeros_like(Wxh), np.zeros_like(Whh), np.zeros_like(Why)
dbh, dby = np.zeros_like(bh), np.zeros_like(by)
dhnext = np.zeros_like(hs[0])
for t in reversed(xrange(len(inputs))):
  dy = np.copy(ps[t])
  dy[targets[t]] -= 1 # backprop into y
  dWhy += np.dot(dy, hs[t].T)
  dby += dy
  dh = np.dot(Why.T, dy) + dhnext # backprop into h
  dhraw = (1 - hs[t] * hs[t]) * dh # backprop through tanh nonlinearity
  dbh += dhraw
  dWxh += np.dot(dhraw, xs[t].T)
  dWhh += np.dot(dhraw, hs[t-1].T)
  dhnext = np.dot(Whh.T, dhraw)
for dparam in [dWxh, dWhh, dWhy, dbh, dby]:
  np.clip(dparam, -5, 5, out=dparam) # clip to mitigate exploding gradients
return loss, dWxh, dWhh, dWhy, dbh, dby, hs[len(inputs)-1]
```



recall:

오류 역전파 단계에서 25반복을 반대 방향으로 진행하게 될 것이고

소프트맥스, 액티베이션에 대해서 gradient 를 계속 구하게 됨

그리고 마지막에 이 클립에 대한 것은 -5 와 5의 안으로 값이설정하도록 하는 것인데

rnn 은 gradient 가 exploding 된다는 단점이 있기 때문에 클립핑을 진행한다고 함.

자세한 건 잘 모르게승ㅁ..

min-char-rnn.py gist

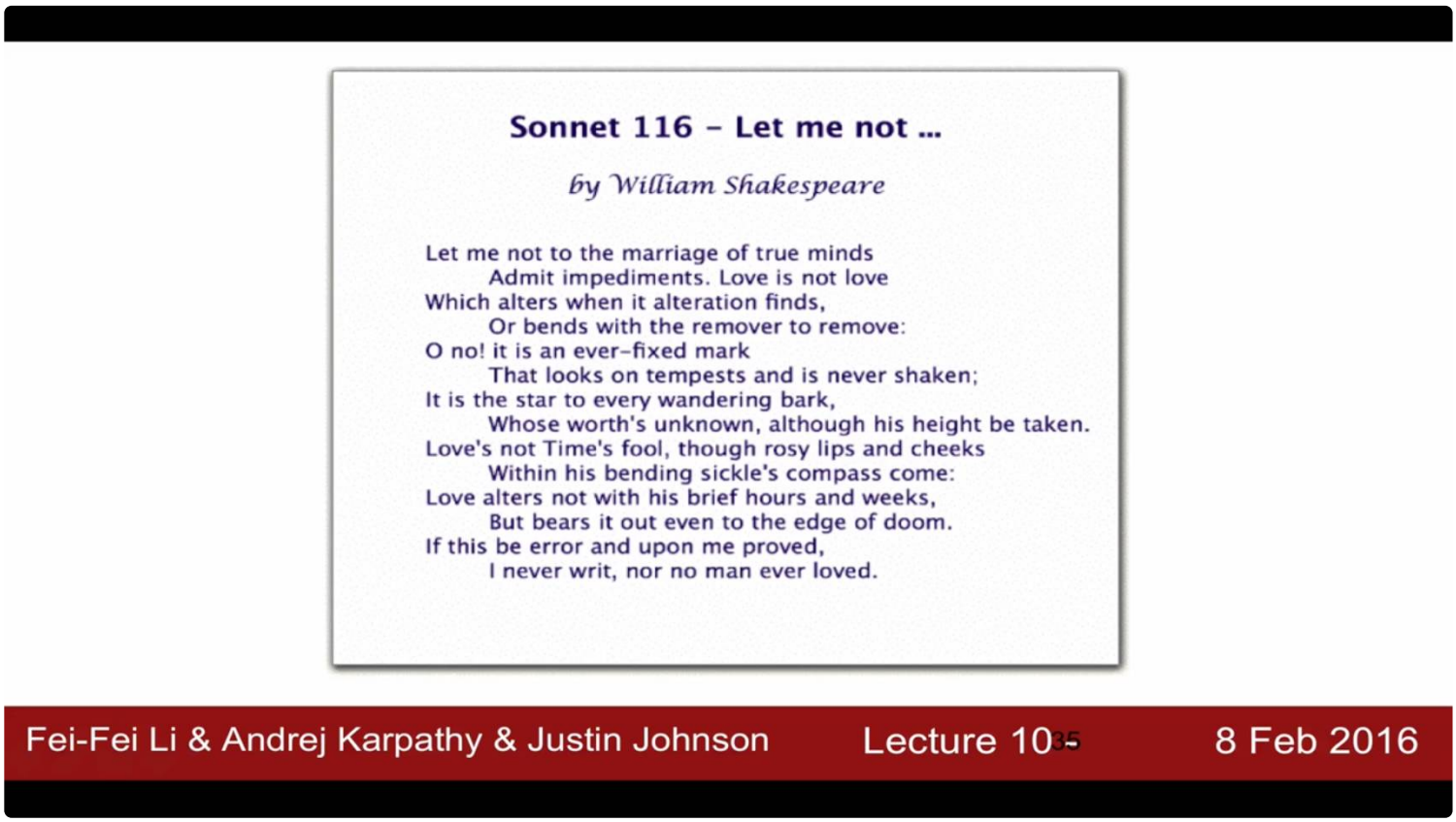
```
def sample(h, seed_ix, n):
  """
  sample a sequence of integers from the model
  h is memory state, seed_ix is seed letter for first time step
  """
  x = np.zeros((vocab_size, 1))
  x[seed_ix] = 1
  ixes = []
  for t in xrange(n):
    h = np.tanh(np.dot(Wxh, x) + np.dot(Whh, h) + bh)
    y = np.dot(Why, h) + by
    p = np.exp(y) / np.sum(np.exp(y))
    ix = np.random.choice(range(vocab_size), p=p.ravel())
    x = np.zeros((vocab_size, 1))
    x[ix] = 1
    ixes.append(ix)
  return ixes
```

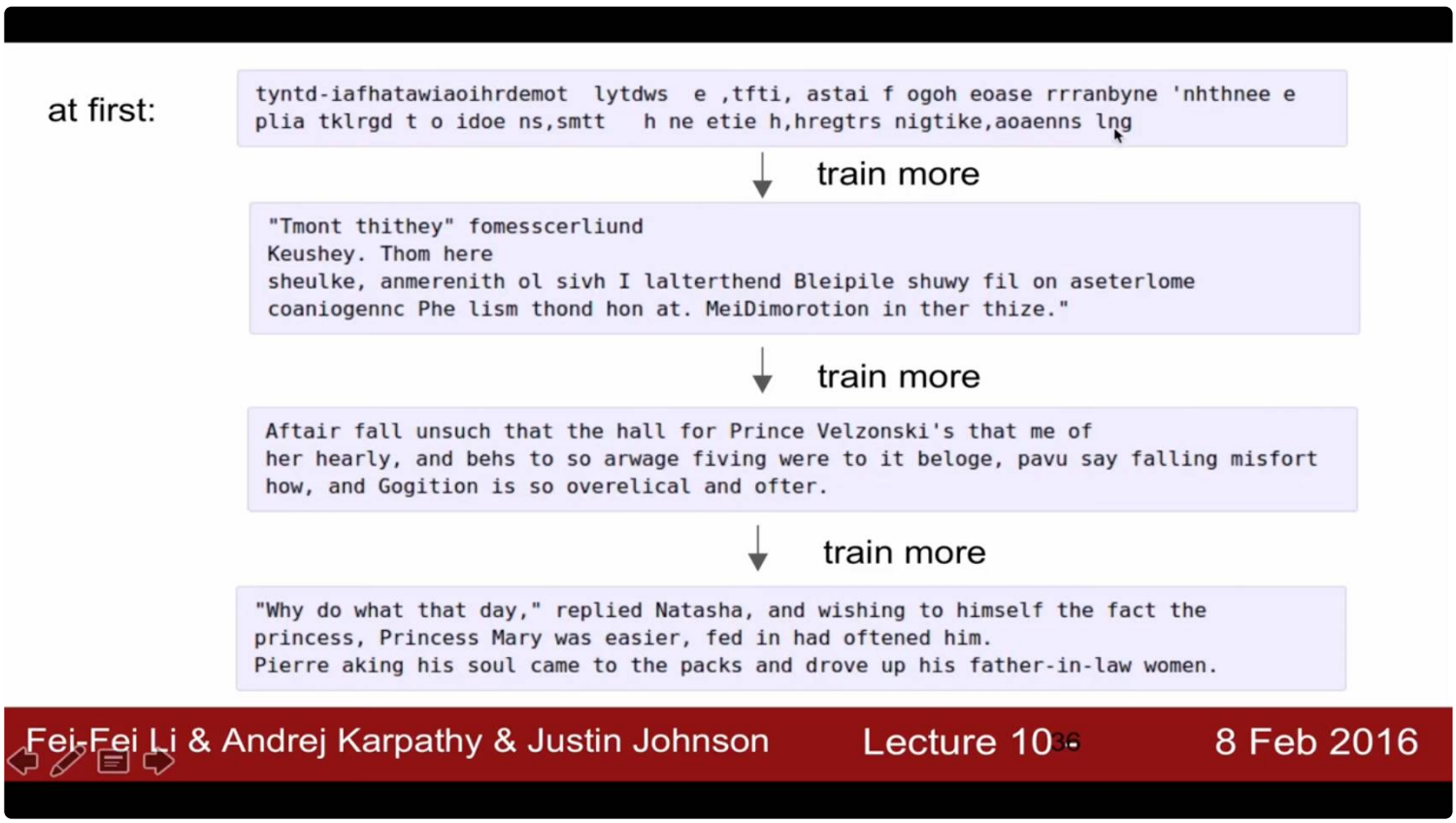
sample 함수는 실제로 학습한 것과 통계에 기반하여서 새로운 텍스트 데이터를 생성하는 함수이다.
새로운원핫 인코딩을 거친 뒤에 리커런스 포뮬라를 이용하여 구해주고

확률을 구해준다

그리고 이 확률을 이용하여 초이스라는 함수를 실행해주고

또 원핫 인코딩을 거친뒤 더해준다.





셰익스피어 뭐시기를 넣어서 학습을 시켰더니 이런식으로 나옴

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```


최종결과

open source textbook on algebraic geometry

The Stacks Project

home about tags explained tag lookup browse search bibliography recent comments blog add slogans

Browse chapters

Part	Chapter	online	TeX source	view pdf
Preliminaries				
	1. Introduction	online	tex	pdf
	2. Conventions	online	tex	pdf
	3. Set Theory	online	tex	pdf
	4. Categories	online	tex	pdf
	5. Topology	online	tex	pdf
	6. Sheaves on Spaces	online	tex	pdf
	7. Sites and Sheaves	online	tex	pdf
	8. Stacks	online	tex	pdf
	9. Fields	online	tex	pdf
	10. Commutative Algebra	online	tex	pdf

Parts

1. Preliminaries
2. Schemes
3. Topics in Scheme Theory
4. Algebraic Spaces
5. Topics in Geometry
6. Deformation Theory
7. Algebraic Stacks
8. Miscellany

Statistics

The Stacks project now consists of

- 455910 lines of code
- 14221 tags (56 inactive tags)
- 2366 sections

Latex source

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariciely in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation ??. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

대수 기하학 교과서를 rnn 에 넣고 돌려본 결과

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y' \to Y' \times_X Y \to X.$$

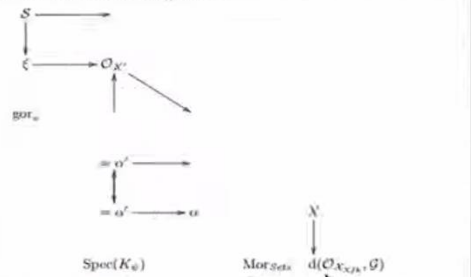
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

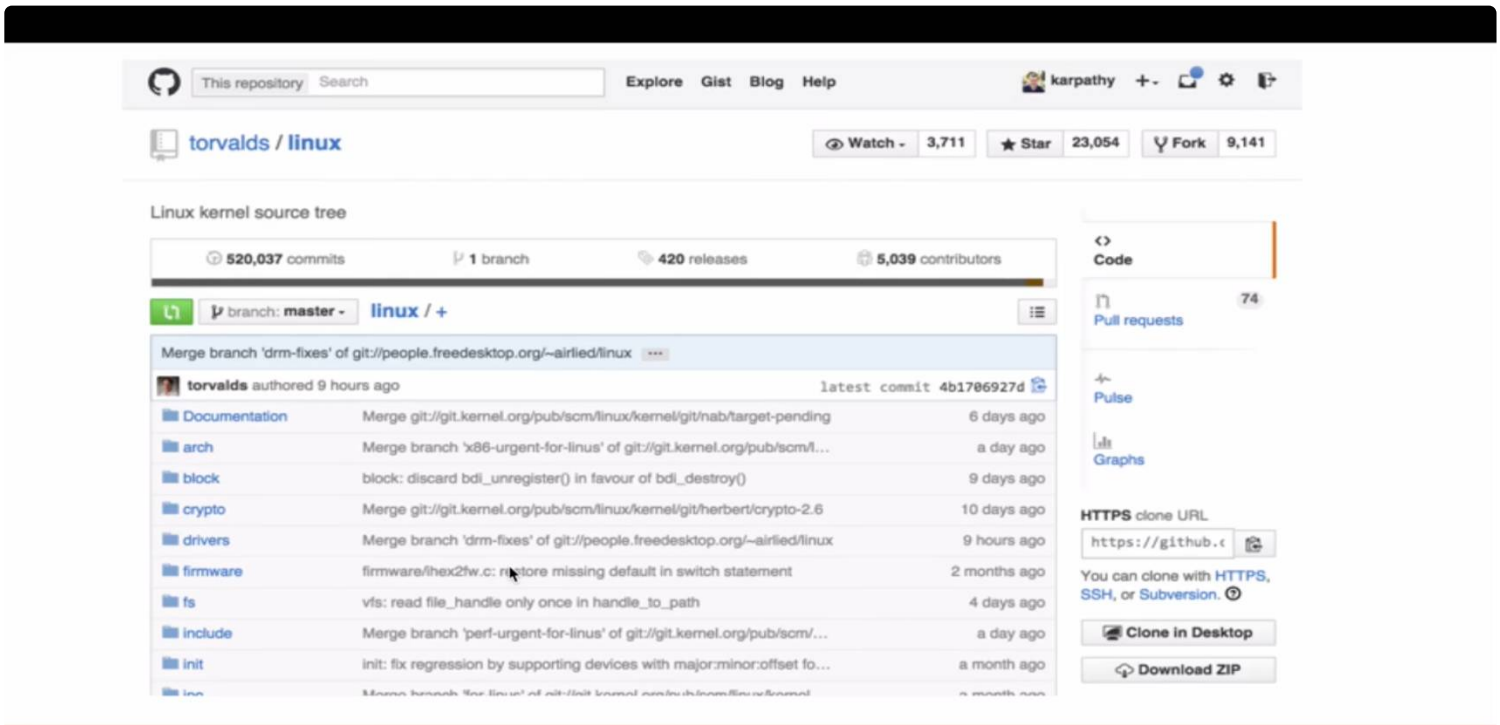
Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{étale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_a}^{\mathcal{E}})$$

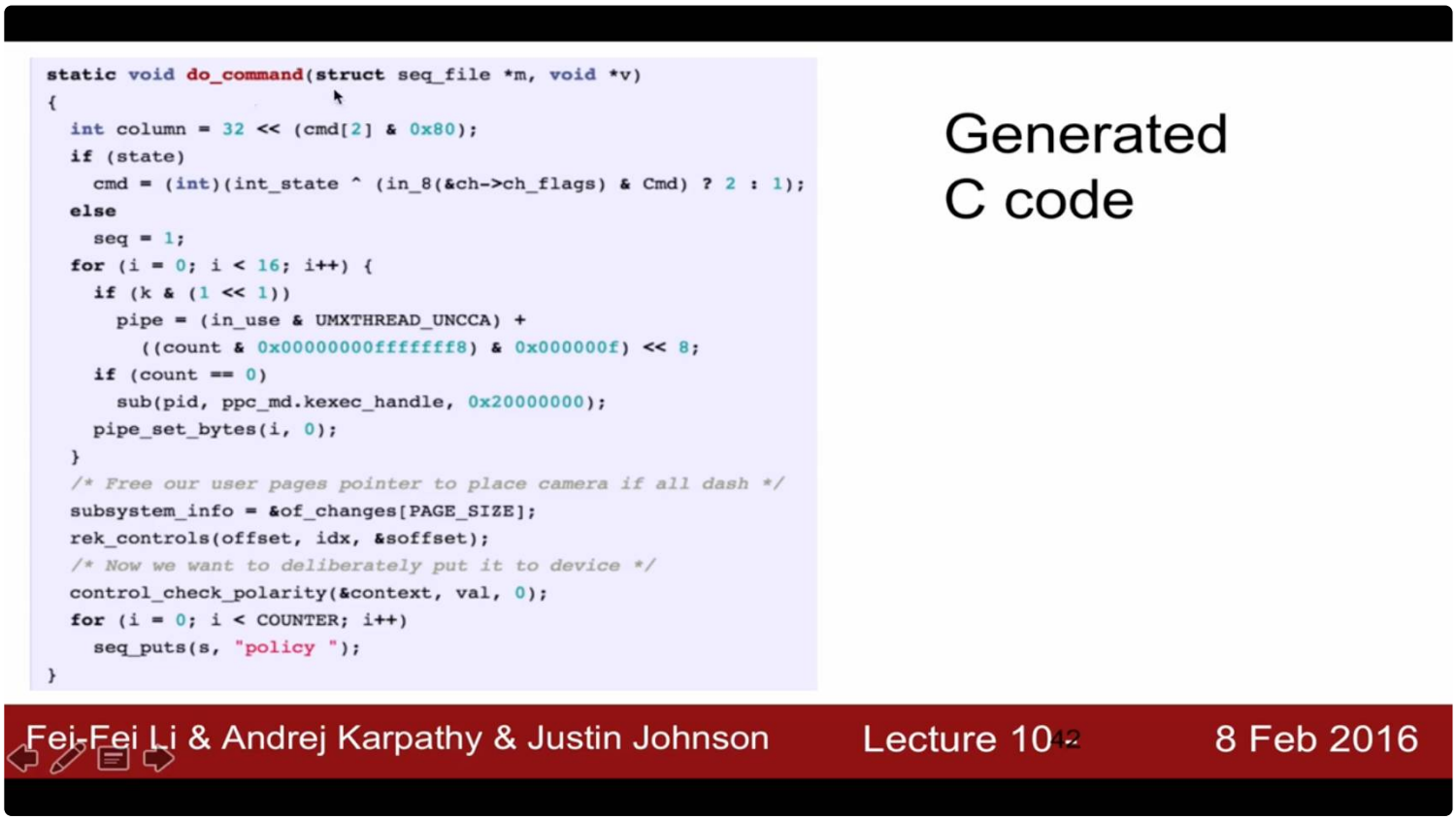
is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

다이어 그램도 알아서 잘 그려준다.



리눅스 토발즈가 개발한 코드...



인자나 인덴테이션 이런것도 훌륭하게 결과를 내놓음

Searching for interpretable cells

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

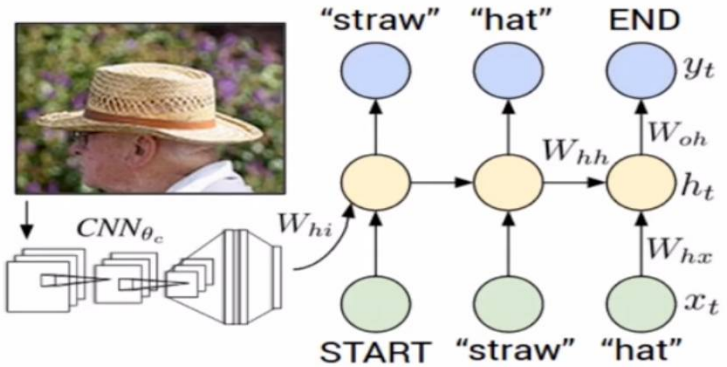
Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

quote detection cell

여기 보듯 시퀀스 길이를 초과한다면 어떻게 될까? 저버

전단계의 히든 스테이트 백터의 상태를 넘겨주고 그리고 이 백자에 대해서만 한계가 되어있지만 generalize 된 어떤 것을 들고와서 작업을 진행하는 것 같다 이런식의 서술

Image Captioning



Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

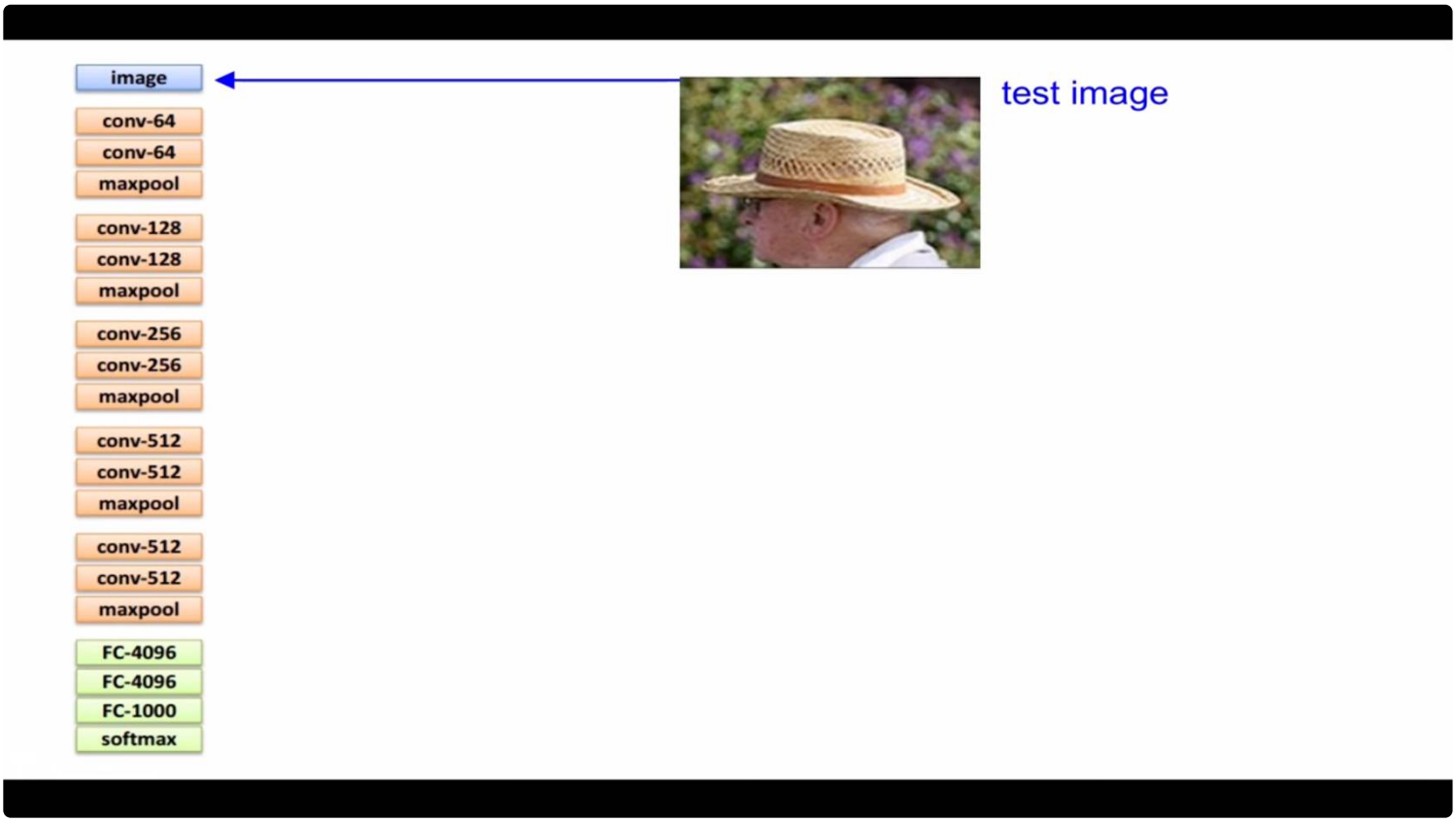
이미지 캡셔닝은 두개로 이루어 지는데

conv 랑 rnn 으로 이루어짐

conv 에서는 이미지를 처리하고

이를 neural network 로 출력해주게 된다.

이런 과정을 좀 살펴 보면

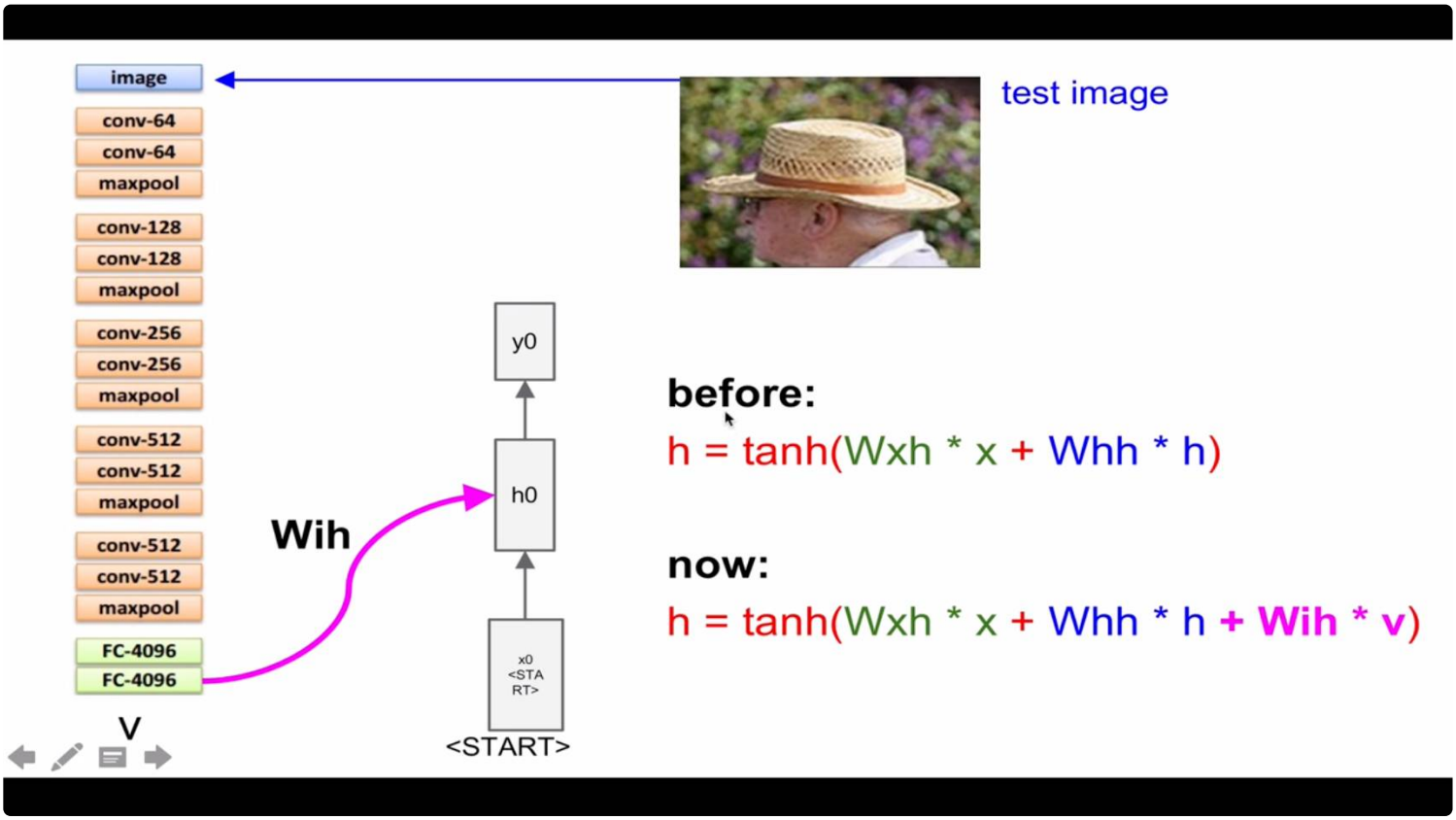


테스트 이미지를 받고 conv net 에 입력을 해준다

fc layer 와 soft max 가 있는데 이 둘을 없애본다

이런 스타트라는 이름의 스패셜한 스타트 백터를 꽂아주고

이게 rnn 에게 이제 시퀀스가 시작된다는 것을 알리는 역할을 하는 것이고



기본적으로 rnn 은 wxh *x + whh * h 를 한 다음에 tanh 를 해준다

이미지 캡셔닝에서는 wxh * x + whh * h + wih * h(이미지 투 히든 * top of conv net 의 값을 곱해준다.)