BITS F464: Machine Learning

34 Sequence Modeling in Neural Network

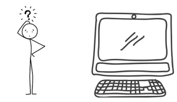
Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

April 19, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

Whether I should go to see a movie?

Suppose you have a system, whom you can ask whether should you go to see a movie or NOT



- The system need to consider many factors such as 1) who are the actors 2) what is IMDB rating of movie 3) your interests in type of movie 4) do you have money to purchase tickets *etc*.
- Everytime the system would give same answer. How many times you can see the same movie?

Basic ML models considers only the current input
(sometime it is useful to consider previous input/output as well)

Application of Sequence Data

There are various places we encounter Sequence Data

Sequence Prediction
Weather Forecasting
Stock Market Prediction
Product Recommendation

Sequence Classification
DNA Sequence Classification
Anomaly Detection
Sentiment Analysis

Sequence Generation
Text Generation
Music Generation
Image Captioning

Sequence-to-Sequence Prediction
Multi-Step Time Series Forecasting
Text Summarization
Language Translation

Sequences

Data is essentially a set of examples (every row represents one)

- Mostly we have fixed number of attributes in an example. However, some interesting applications (such as text, voice, video *etc*) have variable number of points
- These points could depend on each other in complicated way
- We want to do something like
  1. Given a text we want to predict sentiment or next world
     `Agra is a great place, one should go there and visit ________`
  2. Given a voice we want to recognize the speaker
  3. Given a video we want to determine the activity
  4. Given a series (say stock values) predict next one

ISSUE: Basic ML models do **NOT** handle variable number of inputs

Modeling Sequences

There are some ideas to fix the issue (difficulty in using ML models)

- Consider only fixed length.
  (Unable to model long term dependencies)
- Bag of Words: use a vector of length equal to dictionary, and mark/count which words are present
  (since order is not preserved; following lines becomes same)
  `David is good at math but is bad in sciecne`
  `David is bad at math but is good in sciecne`

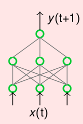
To model sequences we need

1. To deal with variable size input
2. Maintain sequence order
3. Keep track of long term dependencies
4. Share parameters across the sequences

Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network

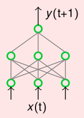


Recurrent Network

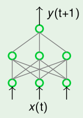
Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network



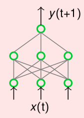
Recurrent Network



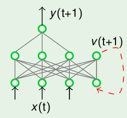
Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network



Recurrent Network

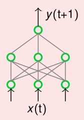


Machine Learning (BITS F464) | M W F (10-11AM) online@BITS-Pilani | Lecture-34(April 19, 2021) | 6 / 24

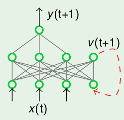
Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network



Recurrent Network



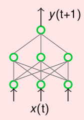
- RNN uses states (called self-state) of the network units available at time t as an input to the other units at time $t+1$

Machine Learning (BITS F464) | M W F (10-11AM) online@BITS-Pilani | Lecture-34(April 19, 2021) | 6 / 24

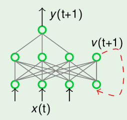
Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network



Recurrent Network



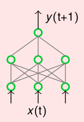
- RNN uses states (called self-state) of the network units available at time t as an input to the other units at time $t+1$
- RNN is suitable for temporal data (like time series)

Machine Learning (BITS F464) | M W F (10-11AM) online@BITS-Pilani | Lecture-34(April 19, 2021) | 6 / 24

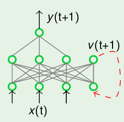
Recurrent Networks (RNN)

Feedforward network cannot capture the dependence of $y(t+1)$ on earlier values of x such as $x(t-1)$

Feedforward Network



Recurrent Network

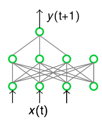


- RNN uses states (called self-state) of the network units available at time t as an input to the other units at time $t+1$
- RNN is suitable for temporal data (like time series)
- Training may involve unfolding and averaging.

Machine Learning (BITS F464) | M W F (10-11AM) online@BITS-Pilani | Lecture-34(April 19, 2021) | 6 / 24

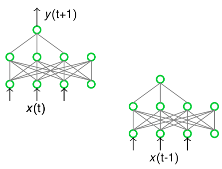
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t+1)$?



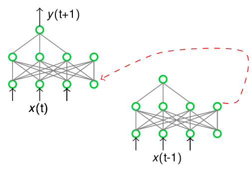
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



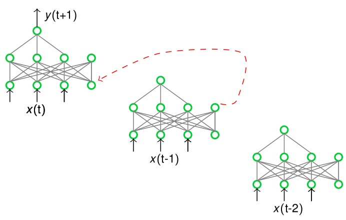
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



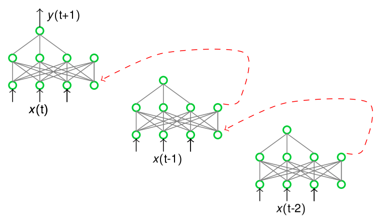
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



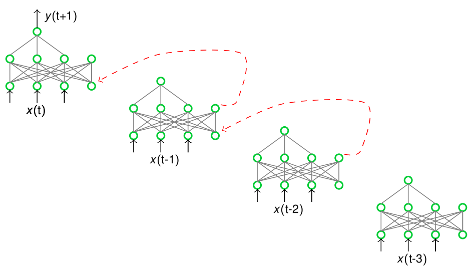
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



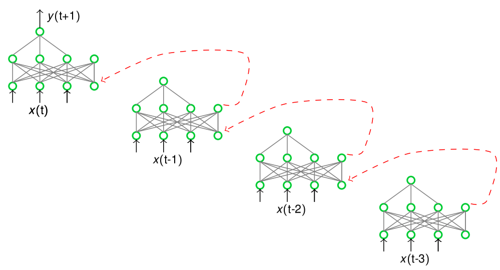
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



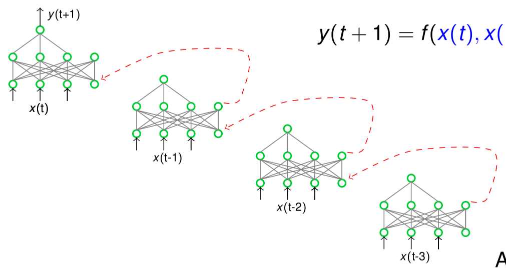
Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t + 1)$?



Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t+1)$?

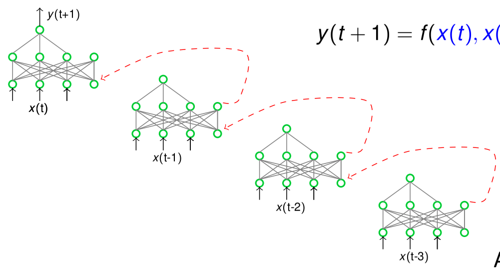


$$y(t+1) = f(x(t), x(t-1), ..., x(0))$$

And so on ...

Unfolding RNN in Time

When input at the time t is provided, what is output at time $(t+1)$?



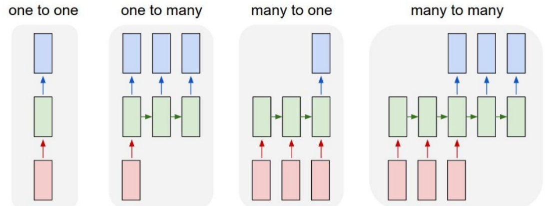
$$y(t+1) = f(x(t), x(t-1), ..., x(0))$$

And so on ...

By this way RNN incorporates history of the network in output

Various arrangements are possible based on need

- `What should be the next world?` **One-to-One**
- `Caption the given image?` **One-to-Many**
- `Segmentation or classification` **Many-to-One**
- `Translate from one language to other` **Many-to-Many**

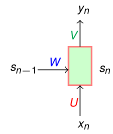


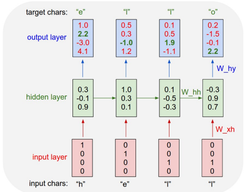
Output may be different for same input

We are optimizing over programs not on functions

Model and an application of RNN [1]

```
rnn = RNN()
y = rnn.step(x)

class RNN:
  def step(self, x):
     self.s = np.tanh(np.dot(self.W, self.s) + np.dot(self.U, x))
     y = np.dot(self.V, self.s)
     return y
```



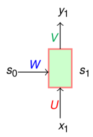


Character-Level Language Models

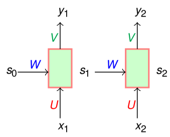
- Given a character what is the next character
- Characters are encoded using one-hot encoding
- Weights need to be adjusted

[1] karpathy.github.io: Minimal character-level Vanilla RNN model

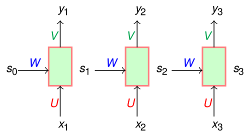
Training: A Simplified Version of RNN



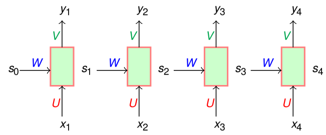
Training: A Simplified Version of RNN



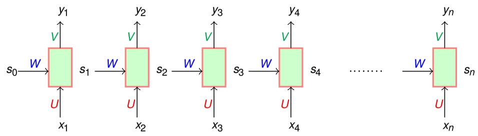
Training: A Simplified Version of RNN



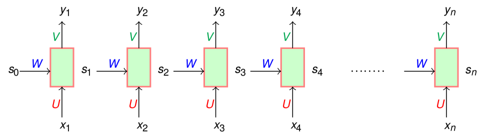
Training: A Simplified Version of RNN



Training: A Simplified Version of RNN

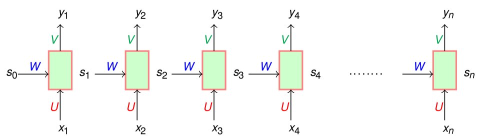


Training: A Simplified Version of RNN



$$y_i = \sigma(V^T s_i) \qquad s_i = \sigma(U^T x_i + W^T s_{i-1})$$

Training: A Simplified Version of RNN

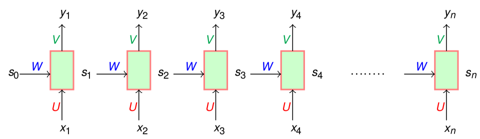


$$y_i = \sigma(V^T s_i) \qquad s_i = \sigma(U^T x_i + W^T s_{i-1})$$

If loss at time t be J_t

Then total loss is $J = \sum_t J_t$

Training: A Simplified Version of RNN



$$y_i = \sigma(V^T s_i) \qquad s_i = \sigma(U^T x_i + W^T s_{i-1})$$

If loss at time t be J_t

Then total loss is $J = \sum_t J_t$

J_t could be something like $-\log$(probability of true output)

How to train RNN?

Training RNN

Backpropogation is used to train a RNN

- Total loss is the summation of J_t at every time step $J = \sum_t J_t$
- To train a parameter we need to find its gradient with respect to loss and shift the parameter in its opposite direction

$$W = W - \alpha \frac{\partial J}{\partial W} = W - \alpha \frac{\partial}{\partial W} \sum_t J_t = W - \alpha \sum_t \frac{\partial J_t}{\partial W}$$

- Consider a single summation term

$$\frac{\partial J_i}{\partial W} = \frac{\partial J_i}{\partial s_i} \times \frac{\partial s_i}{\partial W}$$

- Term $\frac{\partial s_i}{\partial W}$ is complicated. It depends on $s_{i-1}, s_{i-2}, ..., s_1$

Training RNN (contd..)

- As $s_i = \sigma(U^T x_i + W^T s_{i-1})$
  where s_i also depends on W and other previous states $s_{i-2}, ..., s_1$
- Its derivative has **explicit** and **implicit** parts. Explicit part considers other things as constant (we represent as ∂^+)

$$\begin{aligned}
\frac{\partial s_i}{\partial W} &= \frac{\partial^+ s_i}{\partial W} + \frac{\partial s_i}{\partial s_{i-1}} \frac{\partial s_{i-1}}{\partial W} \\
&= \frac{\partial^+ s_i}{\partial W} + \frac{\partial s_i}{\partial s_{i-1}} [\frac{\partial^+ s_{i-1}}{\partial W} + \frac{\partial s_{i-1}}{\partial s_{i-2}} \frac{\partial s_{i-2}}{\partial W}] \\
&= \frac{\partial^+ s_i}{\partial W} + \frac{\partial s_i}{\partial s_{i-1}} \frac{\partial^+ s_{i-1}}{\partial W} + \frac{\partial s_i}{\partial s_{i-1}} \frac{\partial s_{i-1}}{\partial s_{i-2}} [\frac{\partial^+ s_{i-2}}{\partial W} + \frac{\partial s_{i-2}}{\partial s_{i-3}} \frac{\partial s_{i-3}}{\partial W}] \\
&= \sum_{k=1}^{i} \frac{\partial s_i}{\partial s_k} \frac{\partial^+ s_k}{\partial W}
\end{aligned}$$

where we use $\frac{\partial s_i}{\partial s_k}$ as a short form for $\frac{\partial s_i}{\partial s_{i-1}} \frac{\partial s_{i-1}}{\partial s_{i-2}} \frac{\partial s_{i-2}}{\partial s_{i-3}} \cdots \frac{\partial s_{k+2}}{\partial s_{k+1}} \frac{\partial s_{k+1}}{\partial s_k}$

Exploding and Vanishing Gradient

Consider $\frac{\partial s_i}{\partial s_k}$ that is $\frac{\partial s_i}{\partial s_{i-1}} \frac{\partial s_{i-1}}{\partial s_{i-2}} \frac{\partial s_{i-2}}{\partial s_{i-3}} \cdots \frac{\partial s_{k+2}}{\partial s_{k+1}} \frac{\partial s_{k+1}}{\partial s_k} = \Pi_{j=i}^{k+1} \frac{\partial s_j}{\partial s_{j-1}}$

Focus on a single term $\frac{\partial s_j}{\partial s_{j-1}}$
it can be shown that it is upper-bounded by some constant (which depends on W) let's call that value c

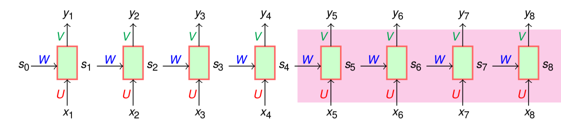
Since $\frac{\partial s_i}{\partial s_k}$ is a multiplication of $i-k$ such terms (and we expect $i-k$ to be large to address long term dependency) therefore the value

$$\frac{\partial s_i}{\partial s_k} \rightarrow \begin{cases} \text{Vanish} & \text{if } c < 1 \\ \text{Explode} & \text{if } c > 1 \end{cases}$$

(0.90→0.81→0.73→0.66→0.59→0.53→0.48→0.43→0.39) (1.50→2.25→3.38→5.06→7.59→11.39→17.09→25.63)

How to Avoid Exploding and Vanishing Gradient

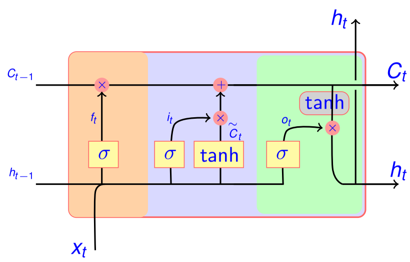
1. **Clipping**
   Try normalizing value of c keeping it some range $T_l \leq c \leq Th$
2. **Truncated Backpropogation**
   At any step look for only for last k timestamps



3. **LSTM**
   Special circuits for handling long term dependencies

LSTM: Long Short Term Memory [2]

LSTM are RNN with 4 special NN circuits for forget, store and output



$f_t = \sigma(W_f[h_{t-1}x_t] + b_f)$

$i_t = \sigma(W_i[h_{t-1}x_t] + b_i)$
$\tilde{C}_t = \tanh(W_C[h_{t-1}x_t] + b_c)$

$\boxed{C_t = f_t * C_{t-1} + i_t * \tilde{C}_t}$

$o_t = \sigma(W_o[h_{t-1}x_t] + b_o)$

$\boxed{h_t = o_t * \tanh(C_t)}$

[2] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Example-01 [1/2]

Five numbers are fed and we want the 2nd one as output.

```
from random import randint
from numpy import array
from numpy import argmax
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense

def generate_sequence(length, n_features):
    return [randint(0, n_features-1) for _ in range(length)]

def one_hot_encode(sequence, n_features):
    encoding = list()
    for value in sequence:
        vector = [0 for _ in range(n_features)]
        vector[value] = 1
        encoding.append(vector)
    return array(encoding)

def one_hot_decode(encoded_seq):
    return [argmax(vector) for vector in encoded_seq]

def generate_example(length, n_features, out_index):
    sequence = generate_sequence(length, n_features)
    encoded = one_hot_encode(sequence, n_features)
    X = encoded.reshape((1, length, n_features))
    y = encoded[out_index].reshape(1, n_features)
    return X, y
```

Example-01 [2/2]

```
### Defining the model ########
length = 5
n_features = 10
out_index = 2
model = Sequential()
model.add(LSTM(25, input_shape=(length, n_features)))
model.add(Dense(n_features, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
model.summary()

##### Fitting the model ############
for i in range(10000):
    X, y = generate_example(length, n_features, out_index)
    model.fit(X, y, epochs=1, verbose=2)

##### Evaluating the model ############
correct = 0
for i in range(100):
    X, y = generate_example(length, n_features, out_index1)
    yhat = model.predict(X)
    if one_hot_decode(yhat) == one_hot_decode(y1):
        correct += 1
print('Accuracy: %f' % ((correct/100)*100.0))
```

The system achieves 100% accuracy

Example-02: Shakespeare like text

All works of Shakespeare (a single 4.4MB file) fed to a 3-layer RNN with 512 hidden nodes on each layer. Output is as below

Original

```
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.
```

RNN output

```
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.
```


Example-02: [1/2] [3]

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
from keras.utils import np_utils

## Data Preprocessing ###
data = open("a.txt").read().lower()
chars = sorted(list(set(data)))
print(chars)
totalChars = len(data)
print(totalChars)
numberOfUniqueChars = len(chars)
print(numberOfUniqueChars)
CharsForids = {char:Id for Id, char in enumerate(chars)}
idsForChars = {Id:char for Id, char in enumerate(chars)}
numberOfCharsToLearn = 100

#### Preprocessing #######
charX = []
y = []
counter = totalChars - numberOfCharsToLearn
for i in range(0, counter, 1):
    theInputChars = data[i:i+numberOfCharsToLearn]
    theOutputChars = data[i + numberOfCharsToLearn]
    charX.append([CharsForids[char] for char in theInputChars])
    y.append(CharsForids[theOutputChars])
X = np.reshape(charX, (len(charX), numberOfCharsToLearn, 1))
X = X/float(numberOfUniqueChars)
y = np_utils.to_categorical(y)
```

[3] https://github.com/sagar448/Keras-Recurrent-Neural-Network-Python/blob/master/README.md

Example-02: [2/2] [4]

```
## Building the RNN model ####
model = Sequential()
model.add(LSTM(256, input_shape=(X.shape[1], X.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(y.shape[1], activation='softmax')) #number of features on the output
model.compile(loss='categorical_crossentropy', optimizer='adam')
model.fit(X, y, epochs=500, batch_size=128)
model.save_weights("Othello.hdf5")

## Generating New Text ###
randomVal = np.random.randint(0, len(charX)-1)
randomStart = charX[randomVal]
print(randomStart)
for i in range(100):
        x = np.reshape(randomStart, (1, len(randomStart), 1))
        x = x/float(numberOfUniqueChars)
        pred = model.predict(x)
        index = np.argmax(pred)
        randomStart.append(index)
        randomStart = randomStart[1: len(randomStart)]
print("".join([idsForChars[value] for value in randomStart]))

## Sample contents from "a.txt" ###
Still questioned me the story of my life
From year to year – the battles, sieges, fortunes
That I have passed.
I ran it through, even from my boyish days
To th' very moment that he bade me tell it.
Wherein I spoke of most disastrous chances,
Of moving accidents by flood and field;
Of hairbreadth scapes i' the' imminent deadly breach;
```

[4] https://github.com/sagar448/Keras-Recurrent-Neural-Network-Python/blob/master/README.md

Example-03: Training with algebraic geometric tex file

Training a multilayer LSTM on *the stacks project* [5] online Latex code (16MB) for Algebraic Geometry book produced

For $\bigoplus_{n=1,\dots,m}$ 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 comparicoly 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 $U \cap U = \coprod_{i=1,\dots,n} U_i$ 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 $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\dots,0}$.

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. □

[5] https://stacks.math.columbia.edu/

Example-04: Training with Linux Source Code [6]

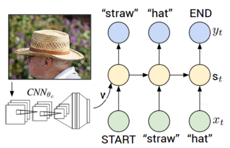
474MB C source code in the *Linux repo on Github* trained on 3-layer LSTMs produced

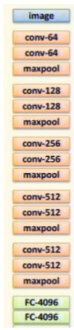
```c
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

```c
/*
 * If this error is set, we will need anything right after that BSD.
 */
static void action_new_function(struct s_stat_info *wb)
{
  unsigned long flags;
  int lel_idx_bit = e->edd, *sys & ~((unsigned long) *FIRST_COMPAT);
  buf[0] = 0xFFFFFFFF & (bit << 4);
  min(inc, slist->bytes);
  printk(KERN_WARNING "Memory allocated %02x/%02x, "
    "original MLL instead\n"),
    min(min(multi_run - s->len, max) * num_data_in),
    frame_pos, sz + first_seg);
  div_u64_w(val, inb_p);
  spin_unlock(&disk->queue_lock);
  mutex_unlock(&s->sock->mutex);
  mutex_unlock(&func->mutex);
  return disassemble(info->pending_bh);
}

static void num_serial_settings(struct tty_struct *tty)
{
  if (tty == tty)
    disable_single_st_p(dev);
  pci_disable_spool(port);
  return 0;
}

static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

[6] https://github.com/torvalds/linux

Example-05: Image Captioning [7]





Before:

$$s_t = \tanh(U^T x_i + W^T s_{i-1})$$

Now:

$$s_t = \tanh(U^T x_i + W^T s_{i-1} + Z^T v)$$

[7] Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

Thank You!

Thank you very much for your attention!