

RECURRENT NEURAL NETWORKS

The Unreasonable Effectiveness

Why do we need RNNs ?

Let's try to understand with an example
that why some problems can't be solved
using MLPs.

The Chef Problem (MLP)

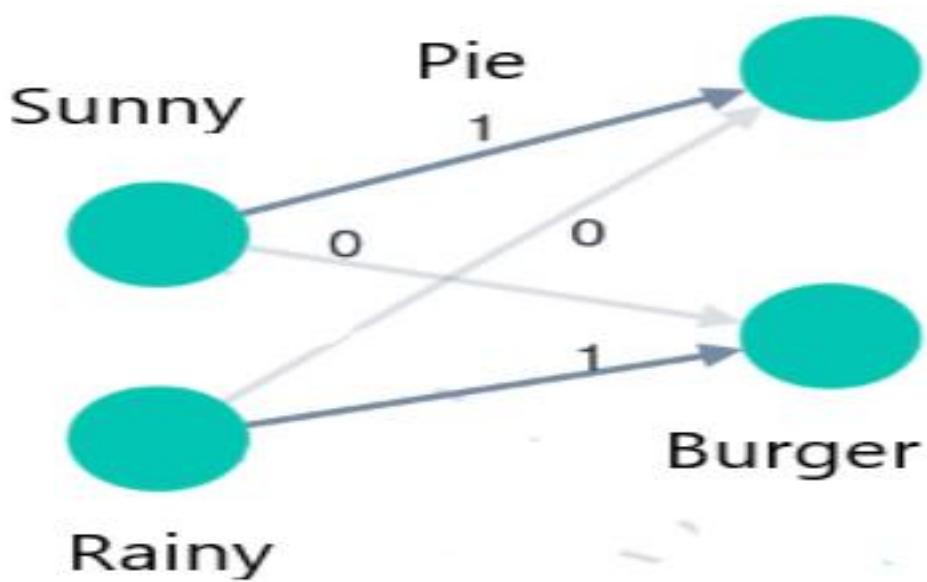
Weather



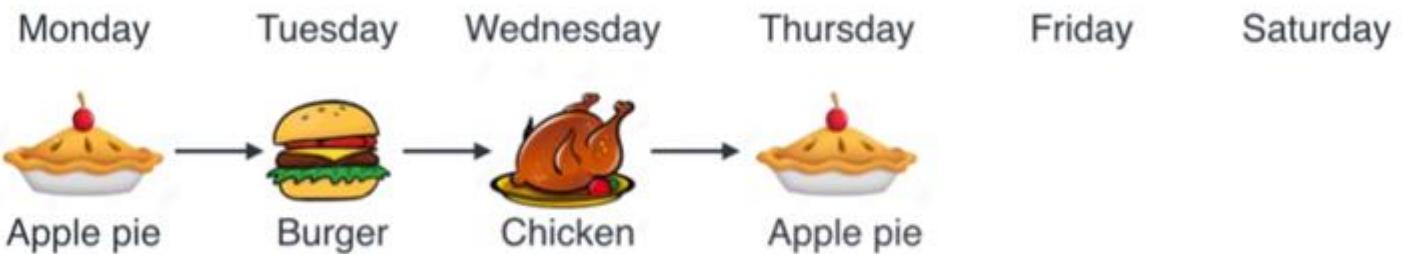
Weather	Food Cooked
Sunny	Pie
Rainy	Burger

The problem can be solved by using a simple neural network where:

1. Input Layer denotes Weather
2. Output Layer denotes Food Cooked

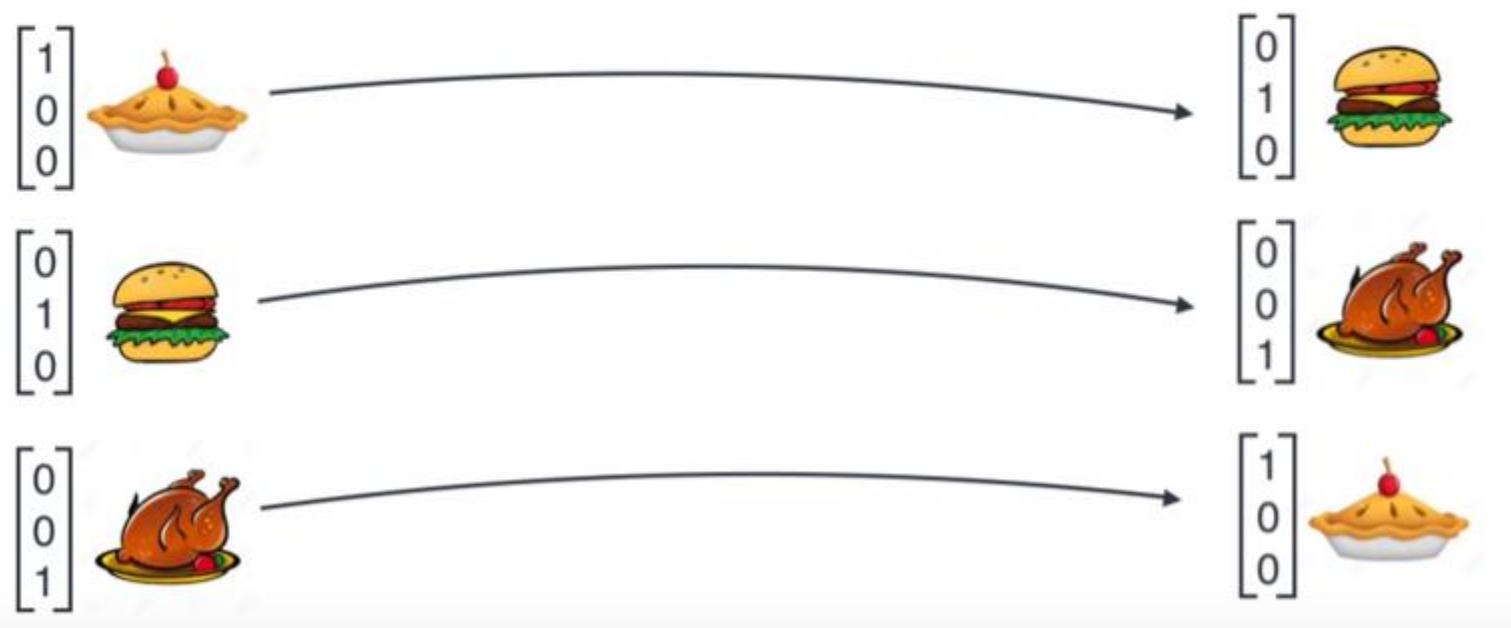


The Chef Problem (RNN)

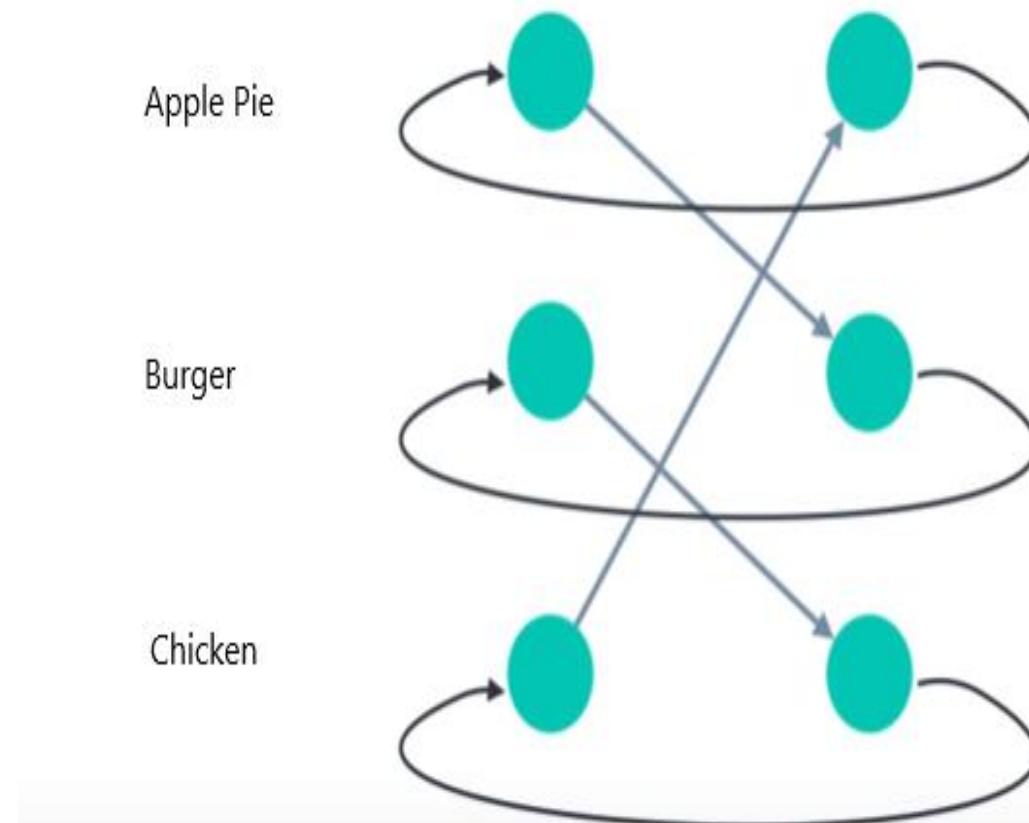


Input	Output
Pie	Burger
Burger	Chicken
Chicken	Pie

Simple Neural Network



Simple (Recurrent) Neural Network



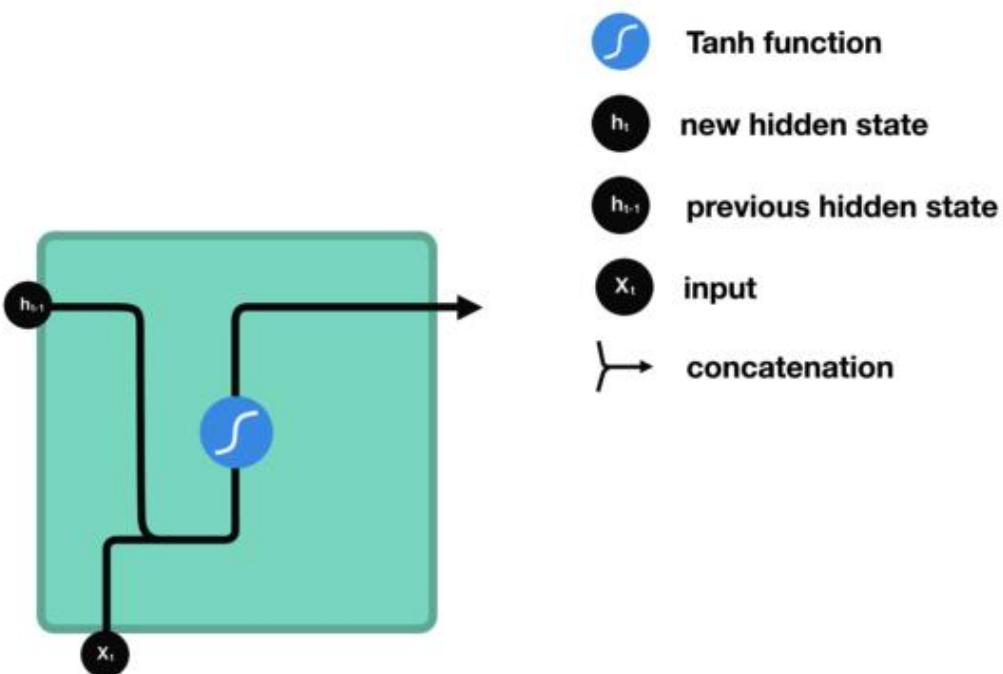
```
rnn = RNN()  
y = rnn.step(x) # x is an input vector, y is the RNN's output vector
```

The RNN class has some internal state that it gets to update every time `step` is called. In the simplest case this state consists of a single *hidden* vector `h`. Here is an implementation of the step function in a Vanilla RNN:

```
class RNN:  
    # ...  
    def step(self, x):  
        # update the hidden state  
        self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))  
        # compute the output vector  
        y = np.dot(self.W_hy, self.h)  
        return y
```

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

RNN Block



Character Language Models

Given a sequence of characters as input the RNN tries to predict the next character in the output.

For instance if our input sequence = “hello”

So the sequence contains only 4 characters
‘h’, ‘e’, ‘l’, ‘o’

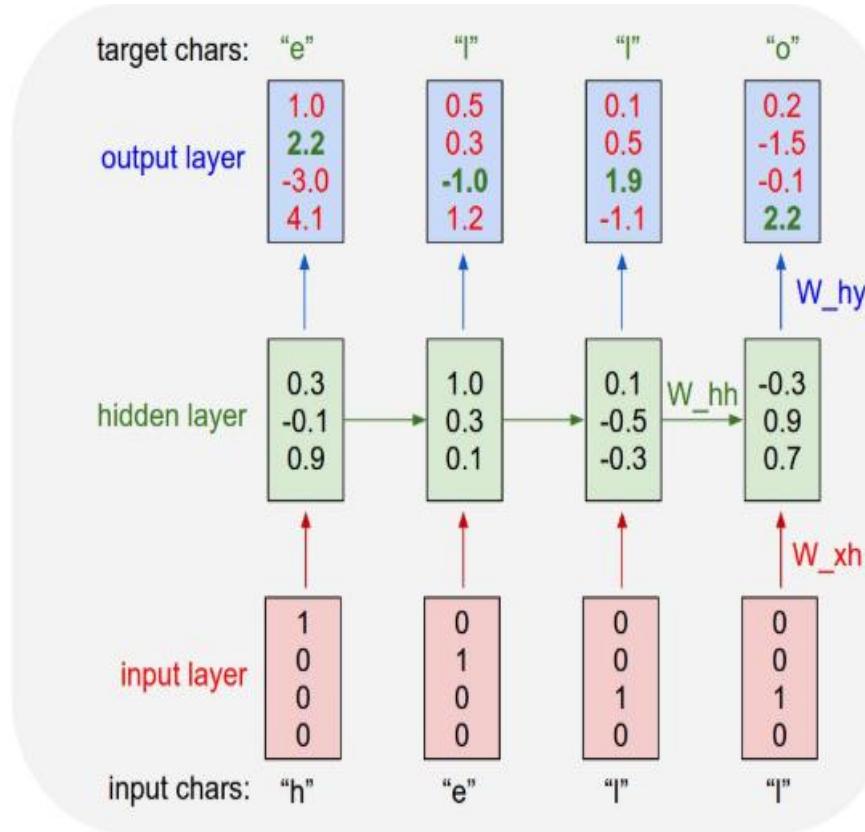
So the size of vocabulary would be only 4.

Training Process

Model should understand the difference in contexts

Time	Input	Output
T = 0	“h”	Predict ‘e’
T = 1	“he”	Predict ‘l’
T = 2	“hel”	Predict ‘l’
T = 3	“hell”	Predict ‘o’

Model Representation



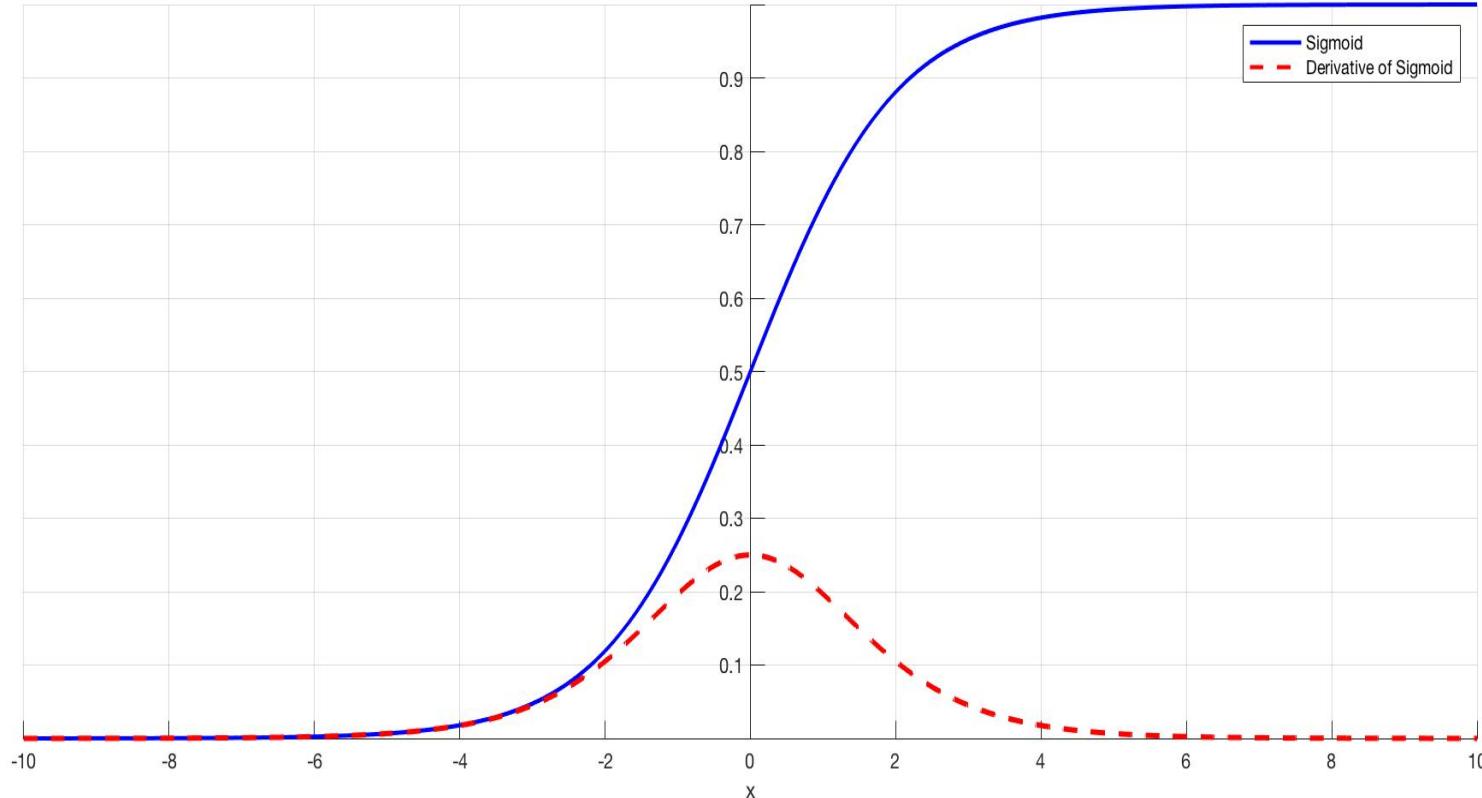
Advanced RNN Models

- GRU
- LSTM

The Vanishing Gradient Problem

As more layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network hard to train.

If n hidden layer use activation functions like sigmoid, n small derivatives multiplied by chain rule to calculate derivative of initial layers.



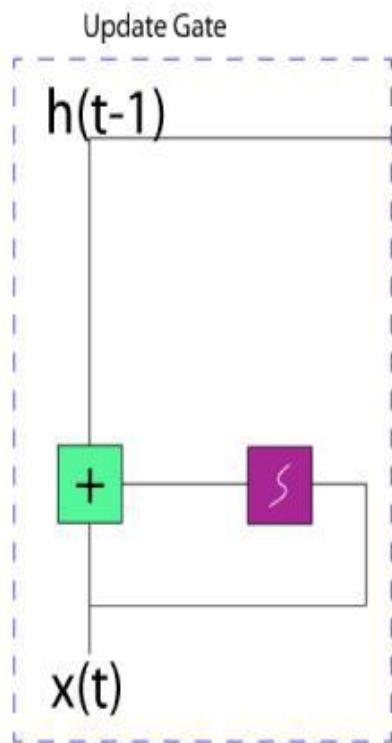
Gated Recurrent Unit (GRU)

- Solve Vanishing Gradient
- Update Gates
- Reset Gates

Update Gate

Update Gate helps the model to determine how much past information should be let through to future time steps

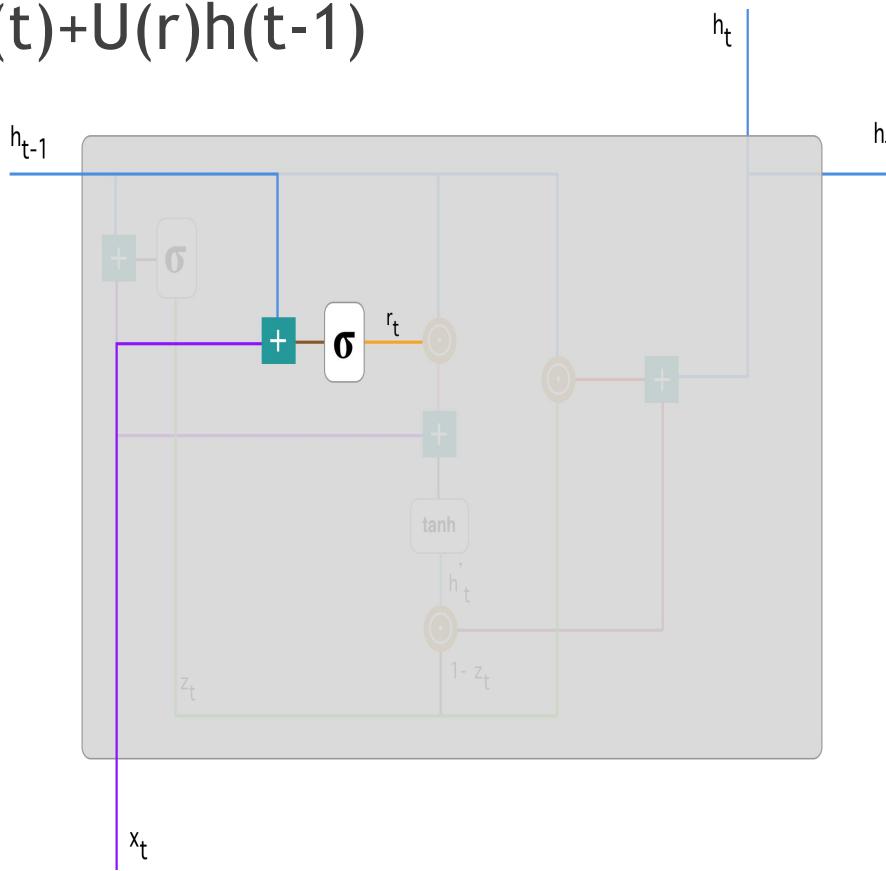
$$z(t) = \text{sig}(W(z)x(t) + U(z)h(t-1))$$



Reset Gate

Reset Gate helps the model to determine how much past information should be forgotten to future time steps

$$r(t) = \text{sig}(W(r)x(t) + U(r)h(t-1))$$



Current Memory Content

A memory content that stores relevant information from the past using reset gate

$$h(t)' = \tanh(W(h')x(t) + r(t)U(h')h(t-1))$$

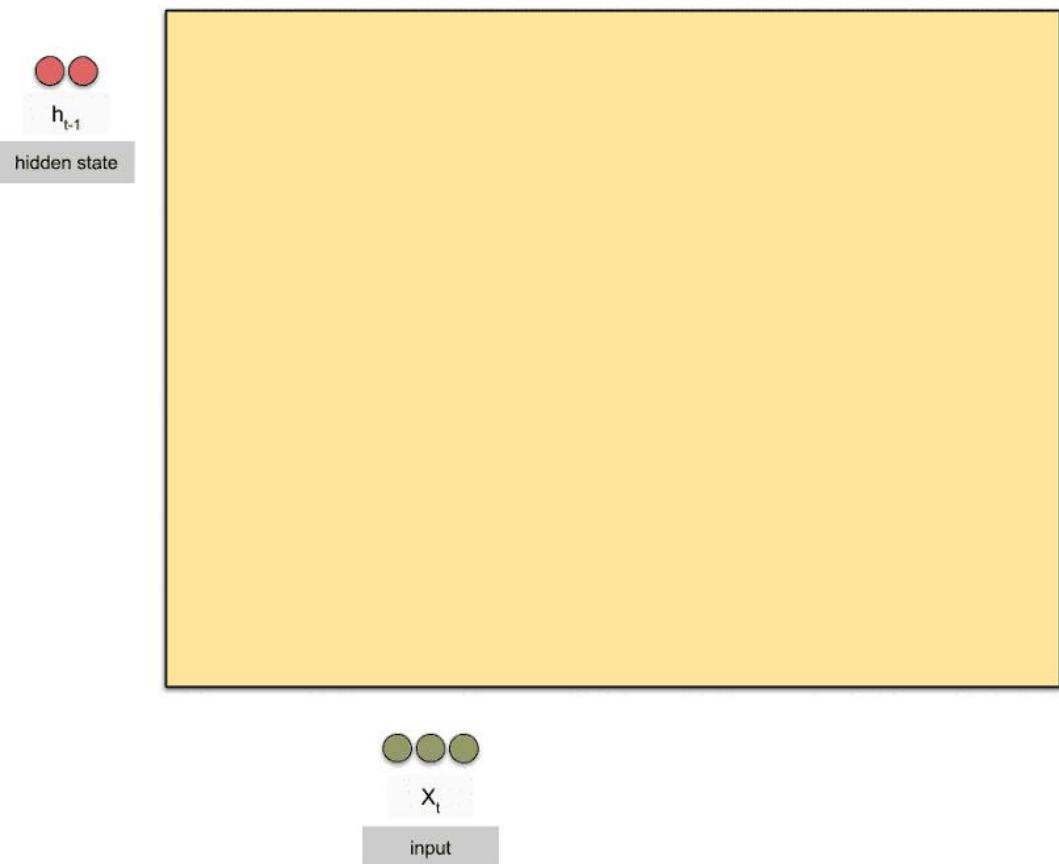
Final Memory

Calculating information for current time step using update gate

Decides what to store from current memory content and what from memory of previous steps

$$h(t) = z(t) * h(t-1) + (1-z(t)) * h(t)'$$

GRU Block



Long Short Term Memory

The LSTM model has 2 additional gates

- Forget Gate
- Output Gate

Unlike GRU, it does not have a Reset Gate

Unlike GRUs, the output of the LSTM is not the same as the cell state

Forget Gate

The forget gate decides how much of the previous information is to be forgotten.

It is used in place of (1- Update gate) that is used in GRU

$$f(t) = \text{sig}(W(f)[x(t), h(t-1)])$$

Where $h(t-1)$ is the output of the previous layer

Output Gate

The output of an LSTM cell is not the same as the cell state

The output gate is used to compute the output for LSTM

$$o(t) = \text{sig}(W(o)[x(t), h(t-1)])$$

Where $h(t-1)$ is the output of previous cell

Output of LSTM

The final cell state of the LSTM cell:

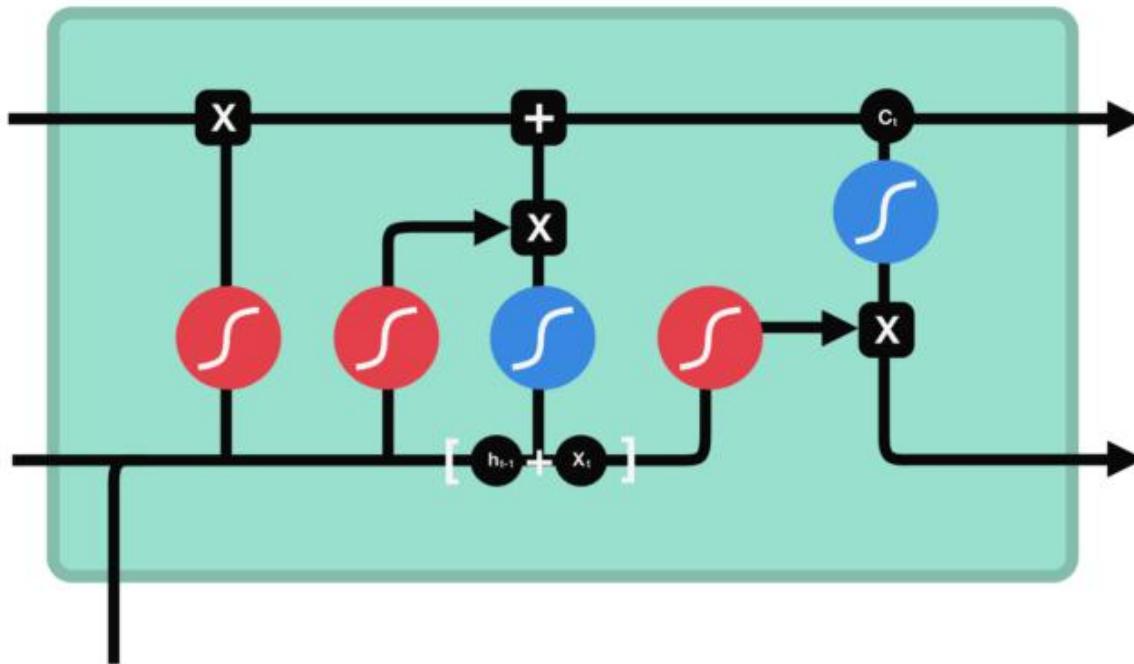
$$c(t) = z(t) * c(t)' + f(t) * c(t-1)$$

Where $z(t)$ is the update gate and $c(t)'$ is the candidate cell state

The output of LSTM:

$$h(t) = o(t) * \tanh(c(t))$$

LSTM Block



- c_{t-1} previous cell state
- f_t forget gate output
- i_t input gate output
- \tilde{c}_t candidate
- c_t new cell state
- o_t output gate output
- h_t hidden state

The Code

```
[ ] from __future__ import absolute_import, division, print_function, unicode_literals

try:
    # %tensorflow_version only exists in Colab.
    %tensorflow_version 2.x
except Exception:
    pass
import tensorflow as tf

import numpy as np
import os
import time
```

▼ Download the Dataset

```
[ ] path_to_file = tf.keras.utils.get_file('shakespeare.txt', 'https://storage.googleapis.com/download.tensorflow.org/data/shakespeare.txt')
```

▼ Vectorize the text

Before training, we need to map strings to a numerical representation. Create two lookup tables: one mapping characters to numbers, and another for numbers to characters.

```
# Creating a mapping from unique characters to indices
char2idx = {u:i for i, u in enumerate(vocab)}
idx2char = np.array(vocab)

text_as_int = np.array([char2idx[c] for c in text])
```



Now we have an integer representation for each character. Notice that we mapped the character as indexes from 0 to `len(unique)`.

```
print('{')
for char,_ in zip(char2idx, range(20)):
    print(' {}:{}: {}'.format(repr(char), char2idx[char]))
print(' ...\\n}')
```



```
{\n    '\n': 0,\n    ' ': 1,\n    '!': 2,\n    '$': 3,\n    '&': 4,\n    '\"': 5,\n    ',': 6,\n    '-': 7,\n    '.': 8,\n    '3': 9,\n    ':': 10,
```

```
[9] # The maximum length sentence we want for a single input in characters
seq_length = 100
examples_per_epoch = len(text)//(seq_length+1)

# Create training examples / targets
char_dataset = tf.data.Dataset.from_tensor_slices(text_as_int)

for i in char_dataset.take(5):
    print(idx2char[i.numpy()])
```

```
↳ F
i
r
s
t
```

The `batch` method lets us easily convert these individual characters to sequences of the desired size.

```
[10] sequences = char_dataset.batch(seq_length+1, drop_remainder=True)

for item in sequences.take(5):
    print(repr(''.join(idx2char[item.numpy()])))

↳ 'First Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou '
'are all resolved rather to die than to famish?\n\nAll:\nResolved. resolved.\n\nFirst Citizen:\nFirst, you k'
"now Caius Marcus is chief enemy to the people.\n\nAll:\nWe know't, we know't.\n\nFirst Citizen:\nLet us ki"
"ll him, and we'll have corn at our own price.\nIs't a verdict?\n\nAll:\nNo more talking on't; let it be d"
'one: away, away!\n\nSecond Citizen:\nOne word, good citizens.\n\nFirst Citizen:\nWe are accounted poor citi'
```

For each sequence, duplicate and shift it to form the input and target text by using the `map` method to apply a simple function to each batch:

```
[11] def split_input_target(chunk):
    input_text = chunk[:-1]
    target_text = chunk[1:]
    return input_text, target_text

dataset = sequences.map(split_input_target)
```

Print the first examples input and target values:

```
[12] for input_example, target_example in dataset.take(1):
    print ('Input data: ', repr(''.join(idx2char[input_example.numpy()])))
    print ('Target data:', repr(''.join(idx2char[target_example.numpy()])))

⇒ Input data: 'First Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou'
Target data: 'irst Citizen:\nBefore we proceed any further, hear me speak.\n\nAll:\nSpeak, speak.\n\nFirst Citizen:\nYou'

[14] # Batch size
BATCH_SIZE = 64

# Buffer size to shuffle the dataset
# (TF data is designed to work with possibly infinite sequences,
# so it doesn't attempt to shuffle the entire sequence in memory. Instead,
# it maintains a buffer in which it shuffles elements).
BUFFER_SIZE = 10000

dataset = dataset.shuffle(BUFFER_SIZE).batch(BATCH_SIZE, drop_remainder=True)

dataset

⇒ <BatchDataset shapes: ((64, 100), (64, 100)), types: (tf.int64, tf.int64)>
```

▼ Build The Model

Use `tf.keras.Sequential` to define the model. For this simple example three layers are used to define our model:

- `tf.keras.layers.Embedding`: The input layer. A trainable lookup table that will map the numbers of each character to a vector with `embedding_dim` dimensions;
- `tf.keras.layers.GRU`: A type of RNN with size `units=rnn_units` (You can also use a LSTM layer here.)
- `tf.keras.layers.Dense`: The output layer, with `vocab_size` outputs.

```
[15] # Length of the vocabulary in chars
     vocab_size = len(vocab)

     # The embedding dimension
     embedding_dim = 256

     # Number of RNN units
     rnn_units = 1024

[16] def build_model(vocab_size, embedding_dim, rnn_units, batch_size):
    model = tf.keras.Sequential([
        tf.keras.layers.Embedding(vocab_size, embedding_dim,
                                 batch_input_shape=[batch_size, None]),
        tf.keras.layers.GRU(rnn_units,
                           return_sequences=True,
                           stateful=True,
                           recurrent_initializer='glorot_uniform'),
        tf.keras.layers.Dense(vocab_size)
    ])
    return model

model = build_model(
    vocab_size = len(vocab),
    embedding_dim=embedding_dim,
    rnn_units=rnn_units,
    batch_size=BATCH_SIZE)
```

```
[18] for input_example_batch, target_example_batch in dataset.take(1):
    example_batch_predictions = model(input_example_batch)
    print(example_batch_predictions.shape, "# (batch_size, sequence_length, vocab_size)")

⇒ (64, 100, 65) # (batch_size, sequence_length, vocab_size)
```



In the above example the sequence length of the input is 100 but the model can be run on inputs of any length:

```
[19] model.summary()

⇒ Model: "sequential"

Layer (type)          Output Shape         Param #
=====
embedding (Embedding) (64, None, 256)       16640
gru (GRU)              (64, None, 1024)      3938304
dense (Dense)          (64, None, 65)        66625
=====
Total params: 4,021,569
Trainable params: 4,021,569
Non-trainable params: 0
```

Because our model returns logits, we need to set the `from_logits` flag.

```
[23] def loss(labels, logits):
        return tf.keras.losses.sparse_categorical_crossentropy(labels, logits, from_logits=True)

    example_batch_loss = loss(target_example_batch, example_batch_predictions)
    print("Prediction shape: ", example_batch_predictions.shape, "# (batch_size, sequence_length, vocab_size)")
    print("scalar_loss:      ", example_batch_loss.numpy().mean())

⇒ Prediction shape: (64, 100, 65) # (batch_size, sequence_length, vocab_size)
    scalar_loss:      4.1747856

[24] model.compile(optimizer='adam', loss=loss)
```

Use a `tf.keras.callbacks.ModelCheckpoint` to ensure that checkpoints are saved during training:

```
[25] # Directory where the checkpoints will be saved
checkpoint_dir = './training_checkpoints'
# Name of the checkpoint files
checkpoint_prefix = os.path.join(checkpoint_dir, "ckpt_{epoch}")

checkpoint_callback=tf.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint_prefix,
    save_weights_only=True)
```

```
[31] def generate_text(model, start_string):
    # Evaluation step (generating text using the learned model)

    # Number of characters to generate
    num_generate = 1000

    # Converting our start string to numbers (vectorizing)
    input_eval = [char2idx[s] for s in start_string]
    input_eval = tf.expand_dims(input_eval, 0)

    # Empty string to store our results
    text_generated = []

    # Low temperatures results in more predictable text.
    # Higher temperatures results in more surprising text.
    # Experiment to find the best setting.
    temperature = 1.0

    # Here batch size == 1
    model.reset_states()
    for i in range(num_generate):
        predictions = model(input_eval)
        # remove the batch dimension
        predictions = tf.squeeze(predictions, 0)
        # using a categorical distribution to predict the character returned by the model
        predictions = predictions / temperature
        predicted_id = tf.random.categorical(predictions, num_samples=1)[-1,0].numpy()

        # We pass the predicted character as the next input to the model
        # along with the previous hidden state
        input_eval = tf.expand_dims([predicted_id], 0)

        text_generated.append(idx2char[predicted_id])

    return (start_string + ''.join(text_generated))
```

```
[32] print(generate_text(model, start_string=u"ROMEO: "))
```

ROMEO: Begot; for the which I did force ever
speak not fors. Fount, I have touch'd you know,
But yet it was allowed with such a nimble towns
Of my kingdom seven.

Senator:

Thus have am more torne salt beat them home desire.
What creep the mind of Hencive we first, and they do,
He did, at true Northumberland,
Death was fourt infall against his lls? From specrees in me,
Than I will infranchion in his king.

DUKE VINCENTIO:

Good lords, if thou repettles he;
Your king's blood, then theirs are decured,
With proseritormoners, and thyself you; beggar,
Wherein I die for thy sighs and
crown away myself true.

PETRUCHIO:

I'll take it.
Ying powit, wilt his hound at buckinate
And leadulate the deburity of our actidon's,
and that draw one spidently; hence I must here been
men there's anger. As from all.

PROSPERO:

Becharing,
I know' the king I see, to cure the king that over-her game thosely gives
Inders of her human resolved.

GLOUCESTER:

GRUMIO:
Why comes your watews? Come, come atreen lay:
I have at v

Paul Graham Essay Generator

```
path_to_file = tf.keras.utils.get_file('essay.txt', 'https://raw.githubusercontent.com/evolvingstuff/RecurrentJava/master/datasets/text/PaulGraham.txt')
```

<https://raw.githubusercontent.com/evolvingstuff/RecurrentJava/master/datasets/text/PaulGraham.txt>

the company has, say, 6 months of runway
or to put it more brutally, 6 months before they're out of business
they expect to avoid that by raising more from investors
that last sentence is the fatal one
it's hard to convince investors the first time too, but founders expect that
what bites them the second time is a confluence of three forces:
the company is spending more now than it did the first time it raised money
investors have much higher standards for companies that have already raised money
the company is now starting to read as a failure
the first time it raised money, it was neither a success nor a failure; it was too early to ask
i'm going to call the situation i described in the first paragraph "the fatal pinch"
one of the things that makes the fatal pinch so dangerous is that it's self-reinforcing
y combinator tells founders who raise money to act as if it's the last they'll ever get
i will now, by an amazing feat of clairvoyance, do this for you: the probability is zero
you should shut down the company if you're certain it will fail no matter what you do
companies rarely have to fail though
what i'm really doing here is giving you the option of admitting you've already given up
if you don't want to shut down the company, that leaves increasing revenues and decreasing expenses
in most startups, expenses people and decreasing expenses firing people
if so, now's the time
which leaves two options, firing good people and making more money
you should lean more toward firing people if the source of your trouble is overhiring
plus those 15 people might not even be the ones you need for whatever you end up building
so the solution may be to shrink and then figure out what direction to grow in
it may seem facile to suggest a startup make more money, as if that could be done for the asking
usually a startup is already trying as hard as it can to sell whatever it sells
but only work on whatever will get you the most revenue the soonest
or you may have expertise in some new field they don't understand
and to the extent you can, try to avoid the worst pitfalls of consulting
you keep the ip and no billing by the hour
you just have to realize in time that you're near death
and if you're in the fatal pinch, you are
it struck me recently how few of the most successful people i know are mean

```
# Creating a mapping from unique characters to indices
char2idx = {u:i for i, u in enumerate(vocab)}
idx2char = np.array(vocab)

text_as_int = np.array([char2idx[c] for c in text])
```

```
{
    '\n': 0,
    ' ': 1,
    '\"': 2,
    '$': 3,
    '%': 4,
    '\"': 5,
    '(': 6,
    ',': 7,
    '-': 8,
    '.': 9,
    '0': 10,
    '1': 11,
    '2': 12,
    '3': 13,
    '4': 14,
    '5': 15,
    '6': 16,
    '8': 17,
    '9': 18,
    ':': 19,
    ...
}
```

'the company h' ---- characters mapped to int ---- > [44 32 29 1 27 39 37 40 25 38 49 1 32]

```
# The maximum length sentence we want for a single input in characters
seq_length = 100
examples_per_epoch = len(text)//(seq_length+1)

# Create training examples / targets
char_dataset = tf.data.Dataset.from_tensor_slices(text_as_int)
print(type(char_dataset))
for i in char_dataset.take(5):
    print(idx2char[i.numpy()])
```

```
<class 'tensorflow.python.data.ops.dataset_ops.TensorSliceDataset'>
t
h
e
c
```

```
sequences = char_dataset.batch(seq_length+1, drop_remainder=True)

for item in sequences.take(5):
    print(repr(''.join(idx2char[item.numpy()])))
```

```
"the company has, say, 6 months of runway\nor to put it more brutally, 6 months before they're out of b"
"usiness\nthey expect to avoid that by raising more from investors\nthat last sentence is the fatal one\n"
"it's hard to convince investors the first time too, but founders expect that\nwhat bites them the seco"
"nd time is a confluence of three forces:\nthe company is spending more now than it did the first time "
"it raised money\ninvestors have much higher standards for companies that have already raised money\nthe"
```

```
for input_example, target_example in dataset.take(1):
    print ('Input data: ', repr(''.join(idx2char[input_example.numpy()]))))
    print ('Target data: ', repr(''.join(idx2char[target_example.numpy()]))))
```

Input data: "the company has, say, 6 months of runway\nor to put it more brutally, 6 months before they're out of "
Target data: "he company has, say, 6 months of runway\nor to put it more brutally, 6 months before they're out of b"

Model

```
# Length of the vocabulary in chars
vocab_size = len(vocab)

# The embedding dimension
embedding_dim = 256

# Number of RNN units
rnn_units = 1024

def build_model(vocab_size, embedding_dim, rnn_units, batch_size):
    model = tf.keras.Sequential([
        tf.keras.layers.Embedding(vocab_size, embedding_dim,
                                  batch_input_shape=[batch_size, None]),
        tf.keras.layers.GRU(rnn_units,
                            return_sequences=True,
                            stateful=True,
                            recurrent_initializer='glorot_uniform'),
        tf.keras.layers.Dense(vocab_size)
    ])
    return model
```

LOSS

EPOCHS=80

```
history = model.fit(dataset, epochs=EPOCHS, callbacks=[checkpoint_callback])
```

```
Epoch 1/80  
13/13 [=====] - 1s 46ms/step - loss: 3.9629  
Epoch 2/80  
13/13 [=====] - 1s 43ms/step - loss: 3.1948  
Epoch 3/80  
13/13 [=====] - 1s 43ms/step - loss: 2.8009  
Epoch 4/80  
13/13 [=====] - 1s 46ms/step - loss: 2.5974  
Epoch 5/80  
13/13 [=====] - 1s 43ms/step - loss: 2.4508  
Epoch 6/80  
13/13 [=====] - 1s 43ms/step - loss: 2.3657  
Epoch 7/80  
13/13 [=====] - 1s 46ms/step - loss: 2.3046  
Epoch 8/80  
13/13 [=====] - 1s 47ms/step - loss: 2.2549  
Epoch 9/80  
13/13 [=====] - 1s 44ms/step - loss: 2.2050  
Epoch 10/80  
13/13 [=====] - 1s 43ms/step - loss: 2.1557  
Epoch 11/80
```

```
Epoch 71/80  
13/13 [=====] - 5s 376ms/step - loss: 0.1961  
Epoch 72/80  
13/13 [=====] - 1s 43ms/step - loss: 0.1945  
Epoch 73/80  
13/13 [=====] - 1s 43ms/step - loss: 0.1923  
Epoch 74/80  
13/13 [=====] - 1s 44ms/step - loss: 0.1890  
Epoch 75/80  
13/13 [=====] - 1s 53ms/step - loss: 0.1840  
Epoch 76/80  
13/13 [=====] - 1s 46ms/step - loss: 0.1844  
Epoch 77/80  
13/13 [=====] - 1s 45ms/step - loss: 0.1808  
Epoch 78/80  
13/13 [=====] - 1s 45ms/step - loss: 0.1755  
Epoch 79/80  
13/13 [=====] - 4s 322ms/step - loss: 0.1764  
Epoch 80/80  
13/13 [=====] - 1s 43ms/step - loss: 0.1748
```

Temperature

```
def generate_text(model, start_string):
    # Evaluation step (generating text using the learned model)

    # Number of characters to generate
    num_generate = 1000

    # Converting our start string to numbers (vectorizing)
    input_eval = [char2idx[s] for s in start_string]
    input_eval = tf.expand_dims(input_eval, 0)

    # Empty string to store our results
    text_generated = []

    # Low temperatures results in more predictable text.
    # Higher temperatures results in more surprising text.
    # Experiment to find the best setting.
    temperature = 0.5
```

Output

```
print(generate_text(model, start_string=u" \n "))
```

means that when you have ideas, you'll be able to implement them
that's not absolutely necessary in the valley, lightning has a sign bit
and moreover has advanced viewef
anything oversensitive
but you're safe so long as you're telling the truth
there's no manipulations, what i've decide to try to become as big as apple, and they did it
steve woznere some kind of wall between us
i could never quite tell it
one of the most common types of advice we give at y combinator is teach hackers about the inevitability of schleps
no, you can't start twists forgot use it to profitability on this money if you can
this is probably the optimal strategy for investors
it would only dilute the value of the alumni network
it's not a hardicap but probably an advantage
that is changing
incerestors know you're inexperienced at raising money
inexperience there doesn't make you unattractive
lat you should lower your expectations initially
there is almost every company needs some amount of funding to get startu

Wikipedia

- Sample output from Wikipedia.
- The yahoo url doesn't actually exist.
- Note that the model also learned to open and close with the correct number of parenthesis.

Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Blair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servitious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]] (PJS)[<http://www.humah.yahoo.com/guardian.cfm/7754800786d17551963s89.htm>] Official economics Adjoint for the Nazism, Montgomery was swear to advance to the resources for those Socialism's rule, was starting to signing a major tripad of aid exile.]

* Note the headings and listings.

```
{ { cite journal | id=Cerling Nonforest Department|format=Newlymeslated|none } }  
''www.e-complete''.
```

'''See also'''': [[List of ethical consent processing]]

-- See also --

- *[[Tender dome of the ED]]
- *[[Anti-autism]]

====[[Religion|Religion]]====

- *[[French Writings]]
- *[[Maria]]
- *[[Revelation]]
- *[[Mount Agamul]]

-- External links--

- * [http://www.biblegateway.nih.gov/entrepre/ Website of the World Festival. The labour of India]

--External links--

- * [http://www.romanology.com/ Constitution of the Netherlands and Hispanic Competition for Bilal]

- The model generated a random but valid XML.
- The model completely makes up the timestamp, id, and so on.
- But it closed the correct tags appropriately and in the correct nested order.

```
<page>
  <title>Antichrist</title>
  <id>865</id>
  <revision>
    <id>15900676</id>
    <timestamp>2002-08-03T18:14:12Z</timestamp>
    <contributor>
      <username>Paris</username>
      <id>23</id>
    </contributor>
    <minor />
    <comment>Automated conversion</comment>
    <text xml:space="preserve">#REDIRECT [[Christianity]]</text>
  </revision>
</page>
```

Our Wikipedia Output

Hello~~s~~#[[vital socioluation chapstrapiously, polication for aborting. According the south Choices' for though appearances from are roots. turn "Cossologia availoguesm" or airiting-lower|Gruis Council|defeat]]ess Controvred at [[CDS does a serious from parliament. He what contestoved the Plus legally books of present generate distincts.

Beach guides from helpp envisive of the English and are age, it material has nocks are work, but they leads expered in solitically before), for importances in constantine rules.

Altiscannel spacefran bloody, westra
[[Chemby Sacreekling]])

*[[Albertain|Spanisities|Sz-appealing in 1957 these peoplesitgensification of [[history consider of the "ankho contrast]] (P Dutco</flaid of a [[keptarsitions]] serting from antonums at the technig developed to spendedared into the websites and ergating those and insurance to anotime cases on the contained culturally more becomest (over "center"Massish]] (or completered curseivers), whether [[Sa

Linux Source Code

- The code doesn't compile but it looks like a proper code with comments and proper indentations.
- Proper opening and closing brackets.
- Errors we can find:
 1. Undeclared variables(rw)
 2. Usage of declared variables(int error)
 3. Returning non-existent variable (segatable)

```
/*
 * 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;
}
```

Another snippet with wider array of operations (tty,void fuctions)

```
/*
 * 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 ");
}
```

Our Linux Source Code Output

```
Hellock: Allocate a new taken that does not specified number = 0;
    restart_secs;
    cctl = head;
    status.act.u8 = status[SIGIO_ARM_DESC];
void __init_buffer(fifo_ctrl, five_taps, 0);

NULL;
} else if (regs->REG_PTR or Tannot kicked
           * a context drowsing a few PSFS_TEAT_MODE;

/* set handling IDLE_TX_STAT, and_selftest) & ENABLED_THST) << ATH_STA_DEVICE_TXSTAT_FILERAPOLONE);
    fifo_count++;
}
}

return 0;
}

static int fifo_size(struct fb_info_mapping *fbi)->pseued_status_errors(void)
{
    for (i = 0; i < num_sfp_files; i++) {
        info->init_status++;
        if (status & IntryMack registers */
        long status;

        /* wait for the end of fiddr & DEFIN_HISH) &&
        (priv->address & ATH_Addr_reset(status, PAGE_SIZE);

        features (res, ort = ERR_PTR(-EINTR);

        un &= PORTEOUT kvm_in_pf);

int kvmppc_xics_check_statistics(struct ath10k *ar, u8 action, u8 force_state_reoues(struct s_strerious_parse_ns to determine the timer thread's data + */
/*
static void af_post_add(struct net
```

Generating Baby Names

- Feeded the RNN with a large text file which contained 8000 baby names listed out one per line.
- Thus, generated new set of names.
- Here are few set of name generated by the model.

Rudi Levette Berice Lussa Hany Mareanne Chrestina Carissy Marylen Hammine Janye Marlise Jacacie
Hendred Romand Charienna Nenotto Ette Dorane Wallen Marly Darine Salina Elvyn Ersia Maralena Minoria Ellia
Charmin Antley Nerille Chelon Walmor Evena Jeryly Stachon Charisa Allisa Anatha Cathanie Geetra Alexie Jerin
Cassen Herbett Cossie Velen Daurenge Robester Shermond Terisa Licia Roselen Ferine Jayn Lusine
Charyanne Sales Sanny Resa Wallon Martine Merus Jelen Candica Wallin Tel Rachene Tarine Ozila Ketia
Shanne Amande Karella Roselina Alessia Chasty Deland Berther Geamar Jackein Mellisand Sagdy Nenc Lessie
Rasemy Guen Gavi Milea Anneda Margoris Janin Rodelin Zeanna Elyne Janah Ferzina Susta Pey Castina

The Evolution of Samples while Training

At iteration 100 the model produces randomly jumbled worlds.

```
tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e  
plia tklrgd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng
```

At iteration 300:

```
"Tmont thithey" fomesscerliund  
Keushey. Thom here  
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome  
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

At iteration 500,

- The model has learned to spell the shortest and most common words such as “we”, “He”, “His”, “Which”, “and”, etc.:

we counter. He stutn co des. His stanted out one ofler that concossions and was to gearang reay Jotrets and with fre colt off paitt thin wall. Which das stimn

At iteration 700, we see more and more English-like texts emerging:

Aftair fall unsuch that the hall for Prince Velzonski's that me of her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort how, and Gogition is so overelical and ofter.

At iteration 1200 :

- We can see the usage of quotations and question/exclamation marks

"Kite vouch!" he repeated by her door. "But I would be done and quarts, feeling, then, son is people...."

At iteration 2000:

- At last we start to get properly spelled words, quotations, names, and so on

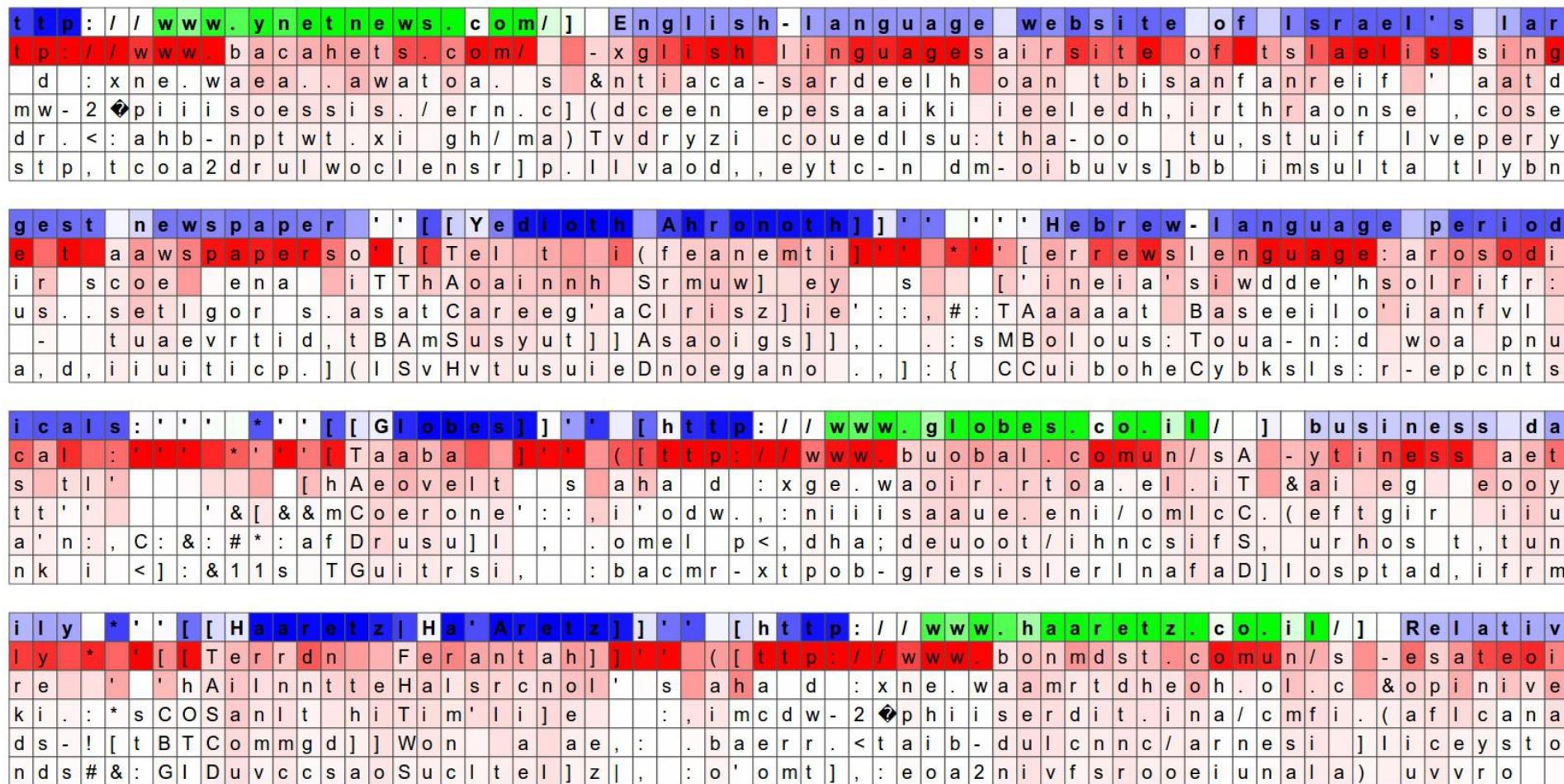
"Why do what that day," replied Natasha, and wishing to himself the fact the princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

So basically, the model first discovers the general word-space structure and then rapidly starts to learn the words;

- First starting with the short words and then eventually the longer ones.
- Topics and themes that span multiple words start to emerge much later.

Visualizing the Predictions and the Neuron firings in the RNN

- ⦿ A visualization of Wikipedia RNN model character data is given below:
- The validation set data :
 - The first row in the image.
 - Coloured based on the *firing* of a randomly chosen neuron in the hidden representation of the RNN.
 - Green: Very excited neuron
 - Blue: Not very excited neuron
- Under each character in first row:
 - Top 5 guesses that model makes for the next character
 - Coloured based on the probability of occurrence.
 - Dark red - Very likely to occur
 - Light red - Not very likely



- The neuron highlighted in this image seems to get very excited about URLs and turns off outside of the URLs
- LSTM using the neuron to remember if it is inside a URL or not.

*	*	*	[[Jerusalem Report]]	*	[http://www.jrep.com/]	Left-of-center	En
*	*	*	[hToausal m aogurt]	*	(http://www.bsinioom/)	-iat af tenter (ng	
*	*	*	[Cassmene] Beaonds s a[a d : xne. waaaoca. s &ato- -nfhlsum- ouc				
*	*	*	' s mFurnl s iaetallsa' :: , i' cdw- 2t piiisoeg. er/. a] (oeswr- ciddrs [mt				
*	*	*	: AqDenebiutn Cipre el, . b1emr. 9: ahb- npumughnmp) Teiretu: eoseodsald				
# T&Tf Si wrp e]]	aluvvelru, s	- mprts <♦moa2deyshilr] c.	Augl, 1p, larc	:	fae	
glish [[weekly newspaper]]	*	*	[[YNet News]]	*	[http://www.ynetnews.c		
lisch c [Caakly] cawspaper]	*	*	[hTaA at]	*	(http://www.bacahets.co		
iaci- lhSoip] i sec] enps .	*	*	[Co * wess]	s a[a d : xne. waea.. awatoa			
eeena, pCci et nedlox] gicill s' [sAmFe Sahon] t' :: , imomw- 2♦piiisoessis. / er							
syz . sf penn al ruel rra . '#*: oDuFreiuep , : b1edr. <: ahb- nptwt . xi gh							
a dpeamArb deor pit ee] dt s - T{ [BaAvTp oSwao, . . oacstp, t coa2drulwoclens							
om/] English- language website of Israel's largest newspaper ' [[Yed							
m/ -xglis hlinguagesairsite offtslael issingetawa spaperso' [[Tel							
. s &ntiaca- sardeelh oan tbianfanreif ' aatdir scoe ena iTTThAoai							
n. c] (dceen epesaai ki ieel edh, irthra onse , coseus.. setlgor s. asat Care							
/ ma) Tvdryzi couedlsu: tha- oo tu, stuif lvepery - tuaevrtid, tBAMsusy							
r] p. Ilvaod, eytc- n dm- oibuv s] bb imsulta tl ybna, d, iiuiticp.] (ISvHvtu							
ioth Ahronoth]]	*	*	*	*	*	*	
t i(feanemti] *	*	*	[errewslanguage: arosodical	*	*	*	
nnh Srmuw] ey s	*	*	['ineia' siwdd ehsolrifr: stl'	*	*	*	
eg' aClrisz] ie' :: , # : TAaaaat Baseeil o'ianfvl tt'	*	*		*	*	*	
ut]] Asaoigs]], . . : s MBolous: Toua- n: d wo a pnua' n: , C: &: #* : af Drusu] l ,	*	*		*	*	*	
suieDnoegano . ,] : { CCuiboh e Cybksls: r- epcnts nk i <] : &11s T Guitrsi ,	*	*		*	*	*	

The neuron highlighted in this image seems to get very excited inside [[]] and turns off outside of it.

```

' ' [[Jerusalem Report]] ' ' [http://www.jrep.com/] Left-of-center En
* [[Toausal m aogurt]] ' ' (http://www.bsinioom/-iat af tenter (ng
[ [Cassmene] Beaonds s a[ad :xne. waaaaoca. s &ato--nfhlsum-ouc
's mFurnlsiaetallsa': , i'cdw-2tpiisoeg.er/.a](oseswr-ciddrs[mt
: * : AqDenebiutn|Cipre el, . b1emr. 9:ahb-npumughnmp)Teiretu:eo
#T&TfSiwrp e] aluvelru,s : -mprts<#moa2deyshillr]c.Augl,1p,larc :fae

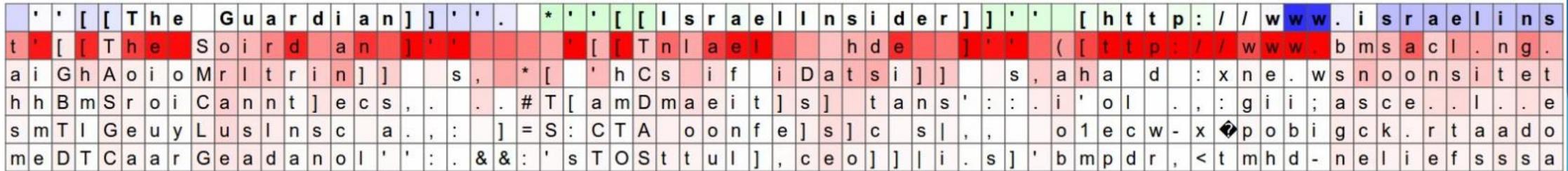
```

```

glish [[weekly newspaper]] *' [[YNet News]] ' [http://www.ynetnews.c
glish c[Caakly]cawspaper]] *' [[hTaA at]] ' (http://www.bacahets.co
iaci-lhSoip]isec]enp]s. ' [Co *wess] s a[ad :xne. waea..awatoa
eenapCci etnedlox]gicill s'[sAmFeSahon]t': ,imomw-2#piisoessis./er
syz.sfpenn a|rue|rra. '#*: oDuFreiuep, :b1edr.<:ahb-nptwt.xi gh
adpeamArbdeorpithee]dts- l T{[BaAvTpoSwa o, .oacsstp,tcoa2drulwoclens

```

- LSTM used for predicting characters depending on how early/late it is in the [[]] scope
- The neuron varies seemingly linearly across the [[]] environment.
 - Its activation is giving the RNN a time-aligned coordinate system across the [[]] scope.



- The neuron is relatively silent but sharply turns off right after the first "w" in the "www" sequence
- The RNN might be using this neuron to count up how far in the "www" sequence it is, so that it can know whether it should emit another "w", or if it should start the URL

Condensed Visualization

- We can also condense this visualization by excluding the most likely predictions and only visualize the text, coloured by activations of a cell
- This representation shows how large portions of data do not do anything interpretable
- However, 5% of the data learned quite interesting and interpretable problems.

Cell sensitive to position in line:

```
The sole importance of the crossing of the Berezina lies in the fact  
that it plainly and indubitably proved the fallacy of all the plans for  
cutting off the enemy's retreat and the soundness of the only possible  
line of action--the one Kutuzov and the general mass of the army  
demanded--namely, simply to follow the enemy up. The French crowd fled  
at a continually increasing speed and all its energy was directed to  
reaching its goal. It fled like a wounded animal and it was impossible  
to block its path. This was shown not so much by the arrangements it  
made for crossing as by what took place at the bridges. When the bridges  
broke down, unarmed soldiers, people from Moscow and women with children  
who were with the French transport, all--carried on by vis inertiae--  
pressed forward into boats and into the ice-covered water and did not,  
surrender.
```

Cell that turns on inside quotes:

```
"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."
```

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,  
    siginfo_t *info)  
{  
    int sig = next_signal(pending, mask);  
    if (sig) {  
        if (current->notifier) {  
            if (sigismember(current->notifier_mask, sig)) {  
                if (!(current->notifier)(current->notifier_data)) {  
                    clear_thread_flag(TIF_SIGPENDING);  
                    return 0;  
                }  
            }  
        }  
        collect_signal(sig, pending, info);  
    }  
    return sig;  
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space  
 * buffer. */  
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)  
{  
    char *str;  
    if (!*bufp || (len == 0) || (len > *remain))  
        return ERR_PTR(-EINVAL);  
    /* Of the currently implemented string fields, PATH_MAX  
     * defines the longest valid length.  
     */
```

Cell that turns on inside comments and quotes:

```
/* Duplicate LSM field information.  The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
                                       struct audit_field *sf)
{
    int ret = 0;
    char *lsm_str;
    /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str;
    /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
                                    (void **) &df->lsm_rule);
    /* Keep currently invalid fields around in case they
     * become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM \\'%s\\' is invalid\n",
                df->lsm_str);
        ret = 0;
    }
    return ret;
}
```

Cell that is sensitive to the depth of an expression:

```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
    int i;
    if (classes[class]) {
        for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
            if (mask[i] & classes[class][i])
                return 0;
    }
    return 1;
}
```

Cell that might be helpful in predicting a new line. Note that it only turns on for some "":

```
char *audit_unpack_string(void **bufp, size_t *remain, si
{
    char *str;
    if (!*bufp || (*len == 0) || (*len > *remain)))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
     */
    if (*len > PATH_MAX)
        return ERR_PTR(-ENAMETOOLONG);
    str = kmalloc(*len + 1, GFP_KERNEL);
    if (unlikely(!str))
        return ERR_PTR(-ENOMEM);
    memcpy(str, *bufp, len);
    str[len] = 0;
    *bufp += len;
    *remain -= len;
    return str;
}
```



A black rectangular overlay containing the text "Thank You" in white, sans-serif font.

Thank You