Practical Deep Learning

micha.codes / fastforwardlabs.com

deep learning can seem *mysterious*

let's find a way to just build a function

Feed Forward Layer

```python
# X.shape == (512,)
# output.shape == (4,)
# weights.shape == (512, 4) == 2048
# biases.shape == (4,)
def feed_forward(activation, X, weights, biases):
    return activation(X @ weights + biases)
```

IE: $f(X) = \sigma(X \times W + b)$

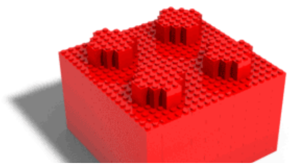
What's so special?

Composable

```
# Just like a Logistic Regression

result = feed_forward(
    softmax,
    X,
    outer_weights,
    outer_biases
)
```

Composable

```
# Just like a Logistic Regression with learned features?

result = feed_forward(
    softmax,
    feed_forward(
        tanh,
        X,
        inner_weights,
        inner_biases
    )
    outer_weights,
    outer_biases
)
```



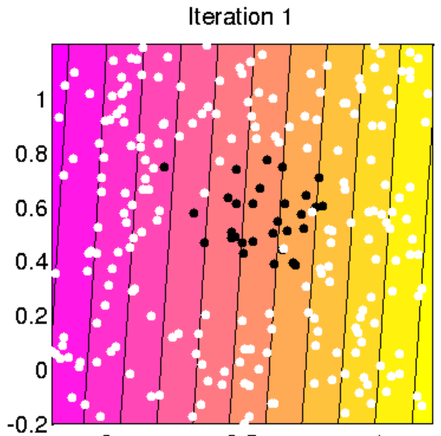
nonlinear

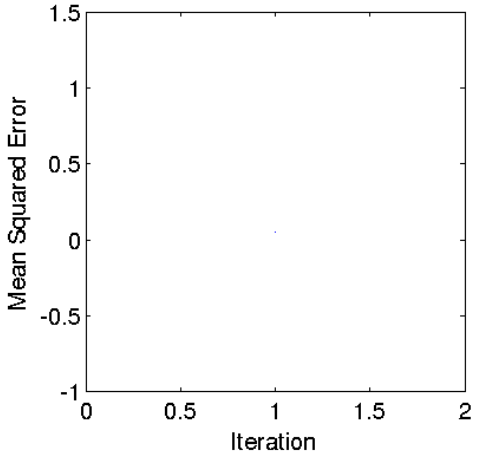
UNIVERSAL APPROXIMATION THEOREM

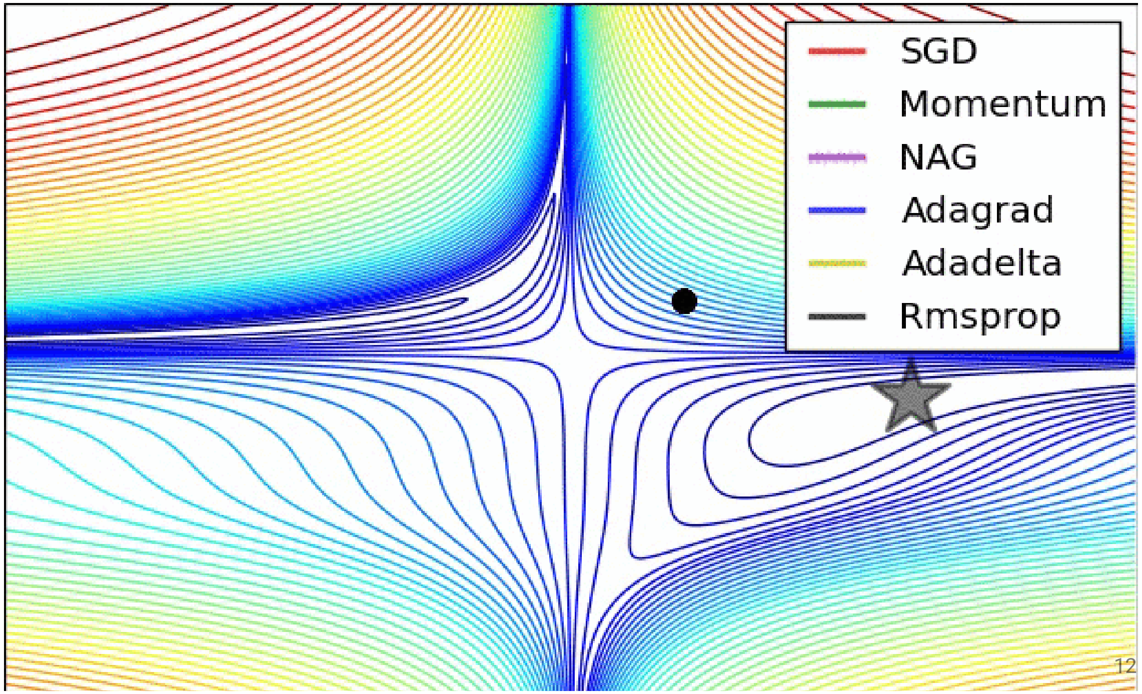
neural networks can approximate arbitrary functions

differentiable SGD

Iteratively learn the values for the weights and biases given training data


Iteration 1
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.5
1


Mean Squared Error
1.5
1
0.5
0
-0.5
-1
0
0.5
1
1.5
2
Iteration

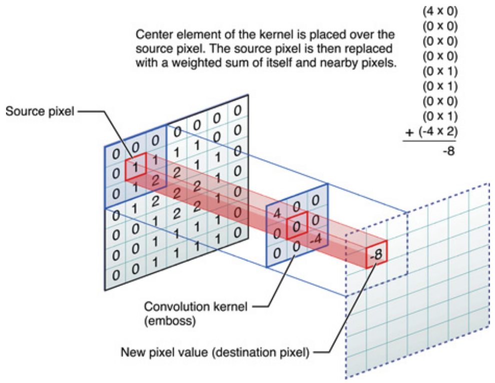

SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop

4GIFs.com

Convolutional Layer

```
import numpy as np
from scipy.signal import convolve

# X.shape == (800, 600, 3)
# filters.shape == (8, 8, 3, 16)
# biases.shape == (3, 16)
# output.shape < (792, 592, 16)
def convnet(activation, X, filters, biases):
    return activation(
        np.stack([convolve(X, f)
                  for f in filter])
        + biases
    )
```

IE: $f(X) = \sigma(X * f + b)$


Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)

Recurrent Layer

```python
# X_sequence.shape == (None, 512)
# output.shape == (None, 4)
# W.shape == (512, 4)
# U.shape == (4, 4)
# biases.shape == (4,)
def RNN(activation, X_sequence, W, U, biases, activation):
    output = None
    for X in X_sequence:
        output = activation(x @ W + output @ U + biases)
        yield output
```

IE: $f_t(X_t) = \sigma\left(X_t \times W + f_{t-1}(X_{t-1}) \times U + b\right)$

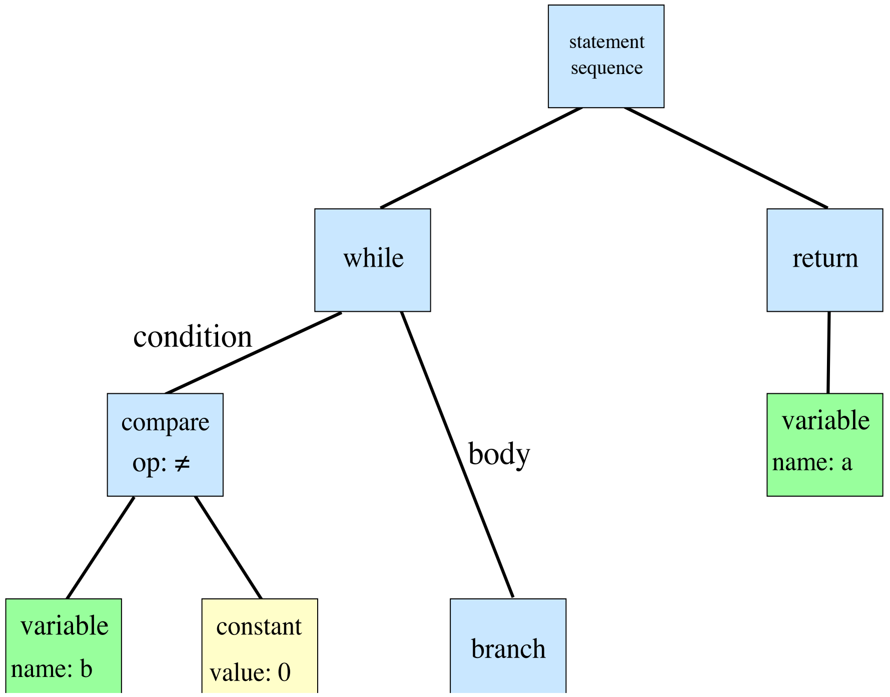
GRU Layer

```python
def GRU(activation_in, activation_out, X_sequence,
        W, U, biases):
    output = None
    for X in X_sequence:
        z = activation_in(W[0] @ X +
                          U[0] @ output +
                          biases[0])
        r = activation_in(W[1] @ X +
                          U[1] @ output +
                          biases[1])
        o = activation_out(W[2] @ x +
                           U[2] @ (r @ output) +
                           biases[2])

        output = z * output + (1 - z) * o
        yield output
```

What about theano/tensorflow/mxnet?

what happens here?

```python
import numpy as np

a = np.random.random(100) - 0.5
a[a < 0] = 10
```


statement sequence
while
return
condition
body
compare
op: ≠
variable
name: a
variable
name: b
constant
value: 0
branch

library	widely used	auto-diff	gpu/cpu	mobile	frontend	models	multi-gpu	speed
numpy								slow
theano								fast
mx-net								fast
tensorflow								slow

Which should I use?

keras makes Deep Learning simple

(http://keras.io/)

```
$ cat ~/.keras/keras.json
{
    "image_dim_ordering": "th",
    "epsilon": 1e-07,
    "floatx": "float32",
    "backend": "theano"
}
```

or

```
$ cat ~/.keras/keras.json
{
    "image_dim_ordering": "tf",
    "epsilon": 1e-07,
    "floatx": "float32",
    "backend": "tensorflow"
}
```

(coming soon... hopefully)

```
$ cat ~/.keras/keras.json
{
    "image_dim_ordering": "mx",
    "epsilon": 1e-07,
    "floatx": "float32",
    "backend": "mxnet"
}
```

simple!

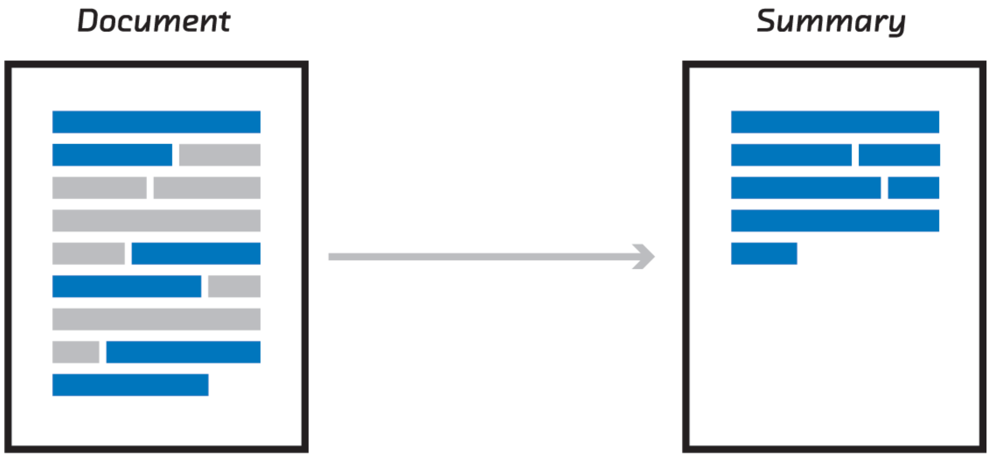
```python
from keras.models import Sequential
from keras.layers.core import Dense

# Same as our Logistic Regression above with:
#     weights_outer.shape = (512, 4)
#     biases_outer.shape = (4,)
model_lr = Sequential()
model_lr.add(Dense(4, activation='softmax', input_shape=[512]))
model_lr.compile('sgd', 'categorical_crossentropy')
model_lr.fit(X, y)
```

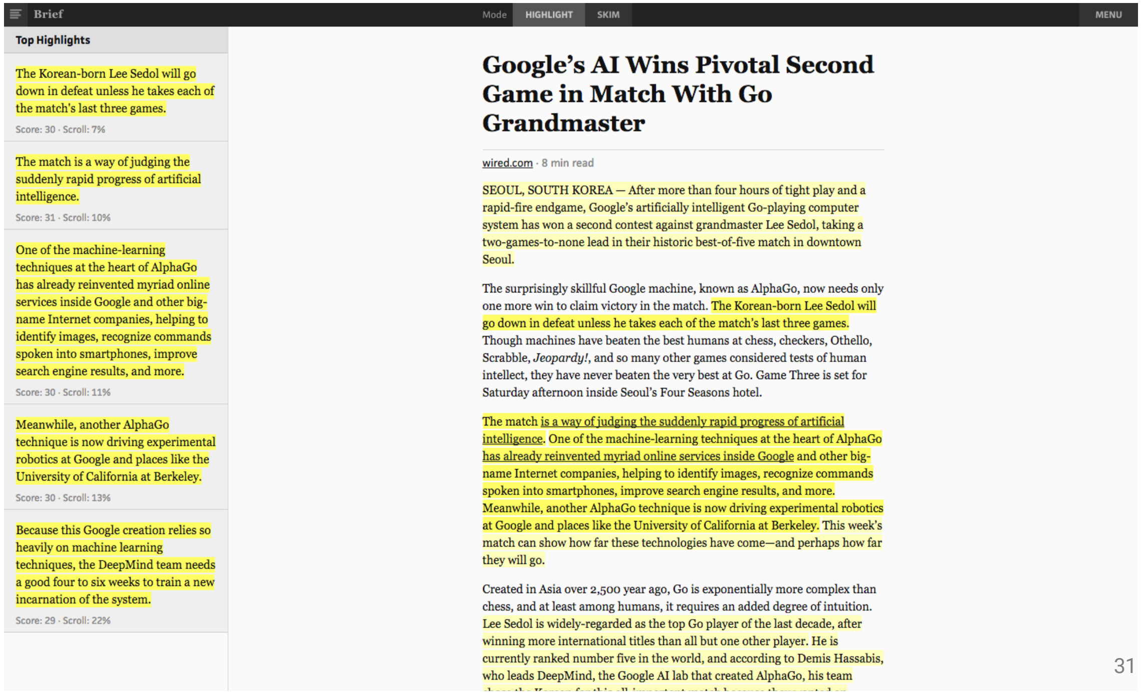
extendible!

```
from keras.models import Sequential
from keras.layers.core import Dense

# Same as our "deep" Logistic Regression
model = Sequential()
model.add(Dense(128, activation='tanh', input_shape=[512]))
model.add(Dense(4, activation='softmax'))
model.compile('sgd', 'categorical_crossentropy')
model.fit(X, y)
```

```
model_lr.summary()
# ____________________________________________________________________________
# Layer (type)              Output Shape    Param #     Connected to
# ============================================================================
# dense_1 (Dense)           (None, 4)       2052        dense_input_1[0][0]
# ============================================================================
# Total params: 2,052
# Trainable params: 2,052
# Non-trainable params: 0
# ____________________________________________________________________________

model.summary()
# _____________________________________________________________________
# Layer (type)       Output Shape    Param #     Connected to
# =====================================================================
# dense_2 (Dense)    (None, 128)     65664       dense_input_2[0][0]
# _____________________________________________________________________
# dense_3 (Dense)    (None, 4)       516         dense_2[0][0]
# =====================================================================
# Total params: 66,180
# Trainable params: 66,180
# Non-trainable params: 0
# _____________________________________________________________________
```

let's build something


Document
Summary



The Korean-born Lee Sedol will go down in defeat unless he takes each of the match's last three games.

Score: 30 · Scroll: 7%

The match is a way of judging the suddenly rapid progress of artificial intelligence.

Score: 31 · Scroll: 10%

One of the machine-learning techniques at the heart of AlphaGo has already reinvented myriad online services inside Google and other big-name Internet companies, helping to identify images, recognize commands spoken into smartphones, improve search engine results, and more.

Score: 30 · Scroll: 11%

Meanwhile, another AlphaGo technique is now driving experimental robotics at Google and places like the University of California at Berkeley.

Score: 30 · Scroll: 13%

Because this Google creation relies so heavily on machine learning techniques, the DeepMind team needs a good four to six weeks to train a new incarnation of the system.

Score: 29 · Scroll: 22%

Google's AI Wins Pivotal Second Game in Match With Go Grandmaster

wired.com · 8 min read

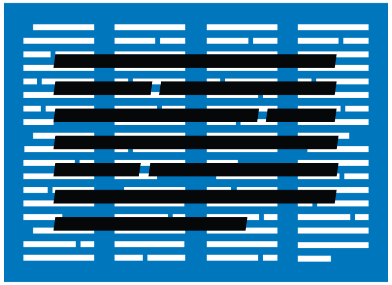
SEOUL, SOUTH KOREA — After more than four hours of tight play and a rapid-fire endgame, Google's artificially intelligent Go-playing computer system has won a second contest against grandmaster Lee Sedol, taking a two-games-to-none lead in their historic best-of-five match in downtown Seoul.

The surprisingly skillful Google machine, known as AlphaGo, now needs only one more win to claim victory in the match. The Korean-born Lee Sedol will go down in defeat unless he takes each of the match's last three games. Though machines have beaten the best humans at chess, checkers, Othello, Scrabble, *Jeopardy!*, and so many other games considered tests of human intellect, they have never beaten the very best at Go. Game Three is set for Saturday afternoon inside Seoul's Four Seasons hotel.

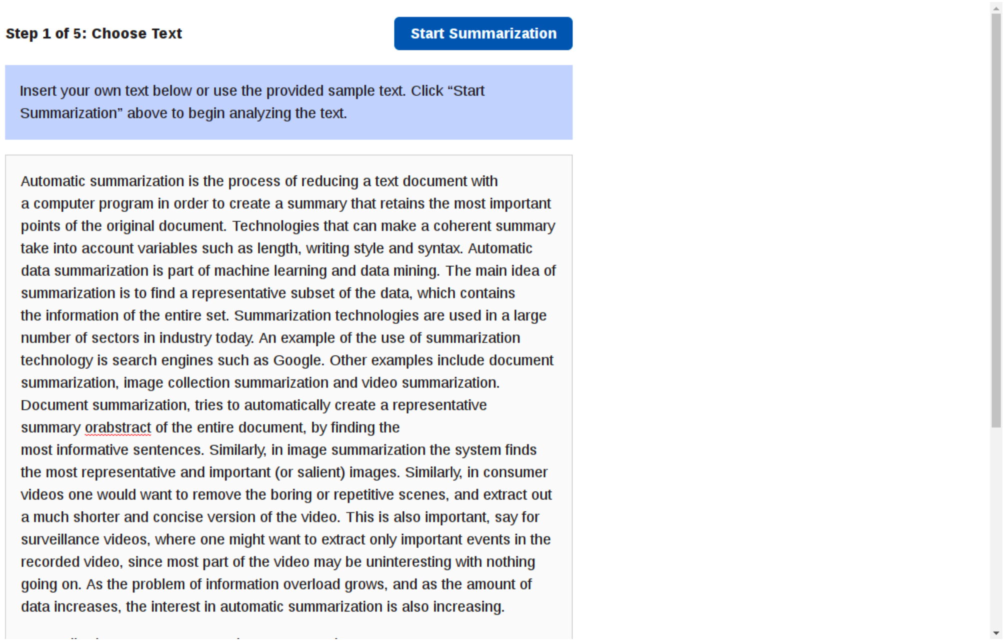
The match is a way of judging the suddenly rapid progress of artificial intelligence. One of the machine-learning techniques at the heart of AlphaGo has already reinvented myriad online services inside Google and other big-name Internet companies, helping to identify images, recognize commands spoken into smartphones, improve search engine results, and more. Meanwhile, another AlphaGo technique is now driving experimental robotics at Google and places like the University of California at Berkeley. This week's match can show how far these technologies have come—and perhaps how far they will go.

Created in Asia over 2,500 year ago, Go is exponentially more complex than chess, and at least among humans, it requires an added degree of intuition. Lee Sedol is widely-regarded as the top Go player of the last decade, after winning more international titles than all but one other player. He is currently ranked number five in the world, and according to Demis Hassabis, who leads DeepMind, the Google AI lab that created AlphaGo, his team

Summarization

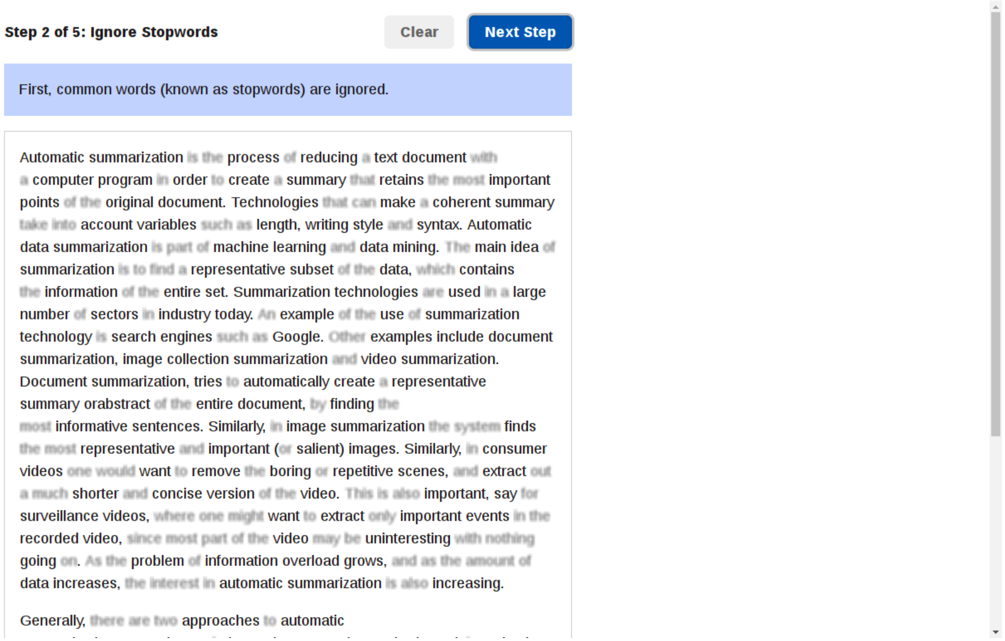


fastforwardlabs.com/luhn/



Insert your own text below or use the provided sample text. Click "Start Summarization" above to begin analyzing the text.

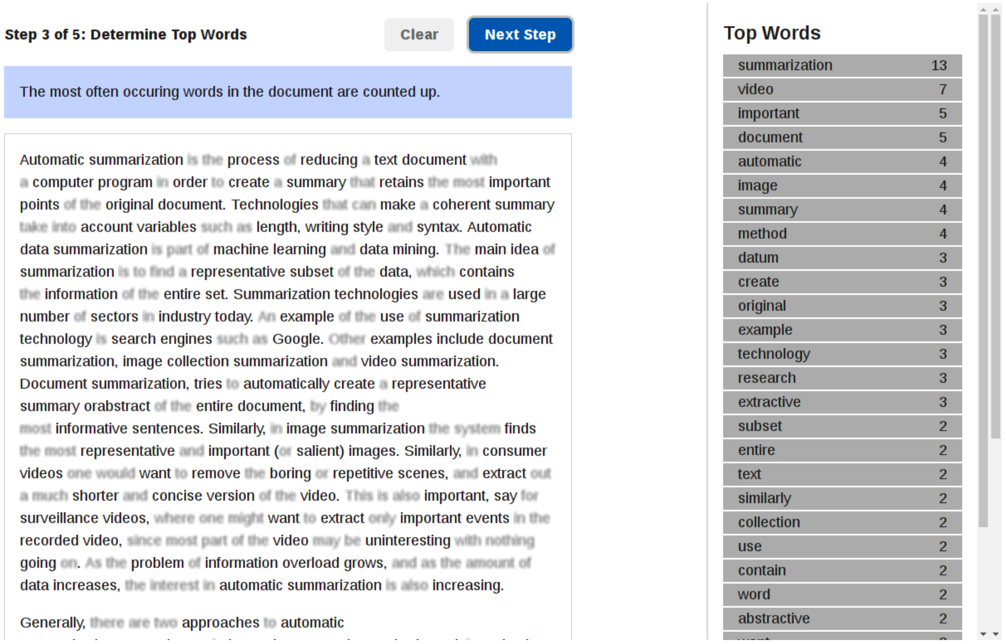
Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.



First, common words (known as stopwords) are ignored.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.

Generally, there are two approaches to automatic



Step 3 of 5: Determine Top Words

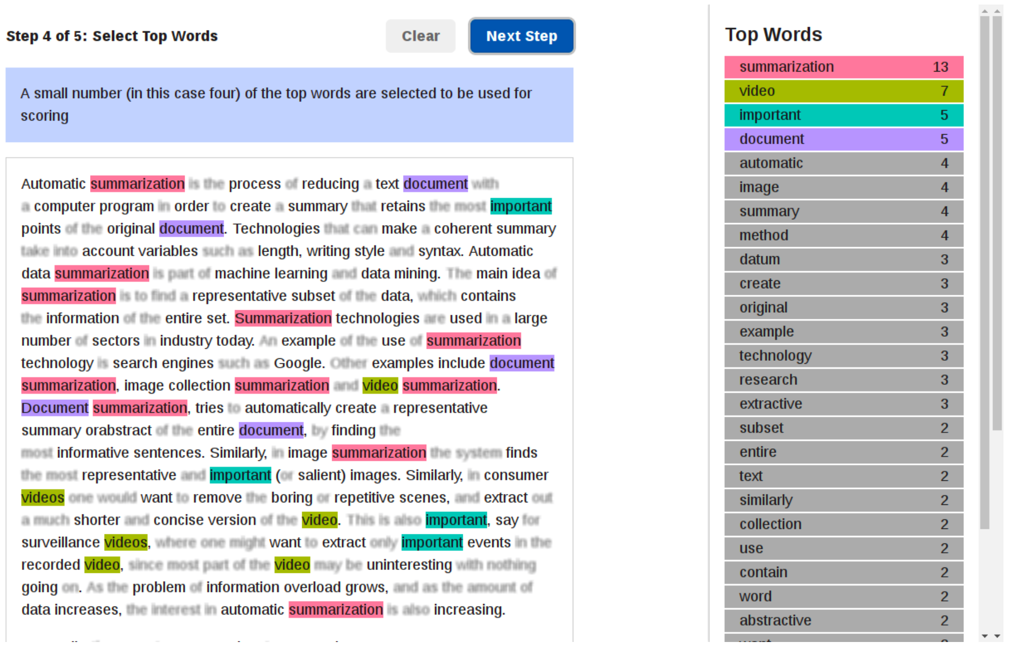
The most often occuring words in the document are counted up.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.

Generally, there are two approaches to automatic

Top Words

summarization	13
video	7
important	5
document	5
automatic	4
image	4
summary	4
method	4
datum	3
create	3
original	3
example	3
technology	3
research	3
extractive	3
subset	2
entire	2
text	2
similarly	2
collection	2
use	2
contain	2
word	2
abstractive	2



Step 4 of 5: Select Top Words

Clear

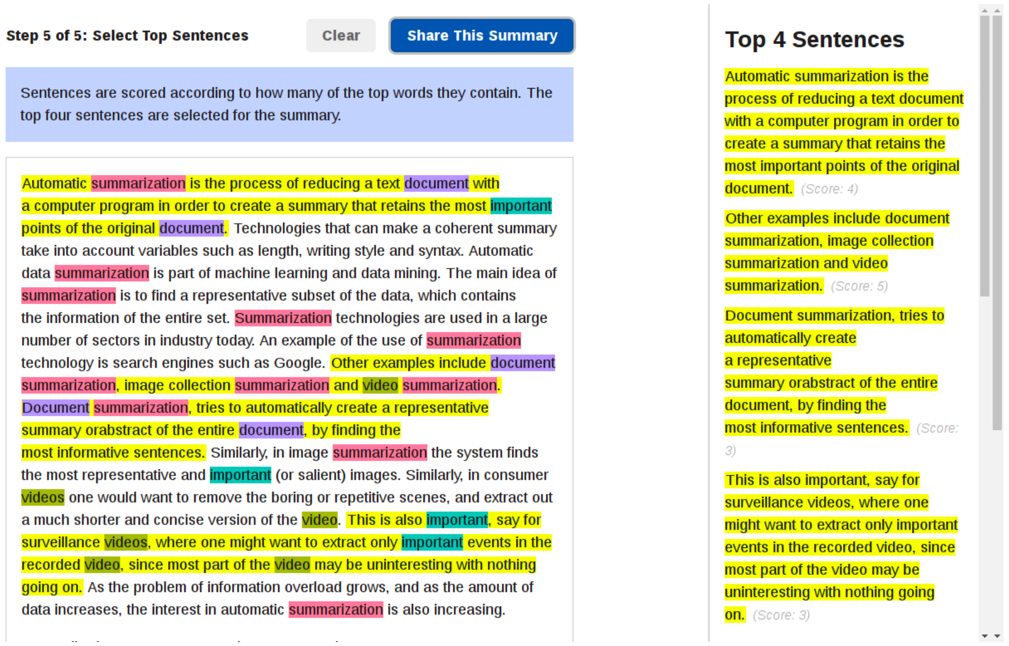
Next Step

A small number (in this case four) of the top words are selected to be used for scoring

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.

Top Words

summarization	13
video	7
important	5
document	5
automatic	4
image	4
summary	4
method	4
datum	3
create	3
original	3
example	3
technology	3
research	3
extractive	3
subset	2
entire	2
text	2
similarly	2
collection	2
use	2
contain	2
word	2
abstractive	2



Sentences are scored according to how many of the top words they contain. The top four sentences are selected for the summary.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. Automatic data summarization is part of machine learning and data mining. The main idea of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. An example of the use of summarization technology is search engines such as Google. Other examples include document summarization, image collection summarization and video summarization. Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. Similarly, in image summarization the system finds the most representative and important (or salient) images. Similarly, in consumer videos one would want to remove the boring or repetitive scenes, and extract out a much shorter and concise version of the video. This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. As the problem of information overload grows, and as the amount of data increases, the interest in automatic summarization is also increasing.

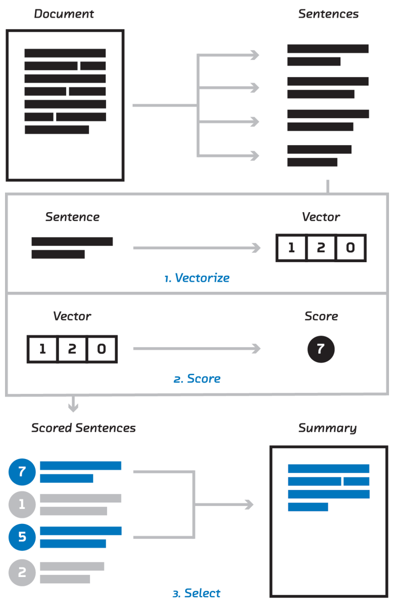
Top 4 Sentences

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. *(Score: 4)*

Other examples include document summarization, image collection summarization and video summarization. *(Score: 5)*

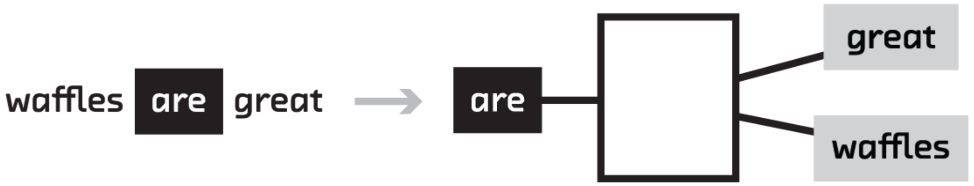
Document summarization, tries to automatically create a representative summary orabstract of the entire document, by finding the most informative sentences. *(Score: 3)*

This is also important, say for surveillance videos, where one might want to extract only important events in the recorded video, since most part of the video may be uninteresting with nothing going on. *(Score: 3)*

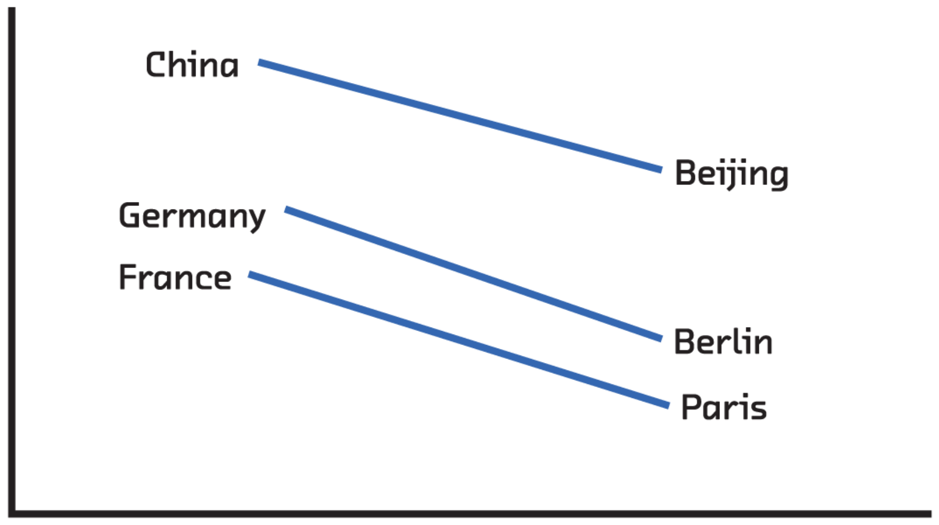

Document
Sentences
Sentence
Vector
1 2 0
1. Vectorize
Vector
1 2 0
Score
7
2. Score
Scored Sentences
7
1
5
2
Summary
3. Select

	cat	sat	mat	dog	bites	man
the cat sat on the mat	1	1	1	0	0	0
cats and dogs	1	0	0	1	0	0
man bites dog	0	0	0	1	1	1
dog bites man	0	0	0	1	1	1

Level 1
Spectrum
English Language
Student Book
ABC
Name
Spectrum



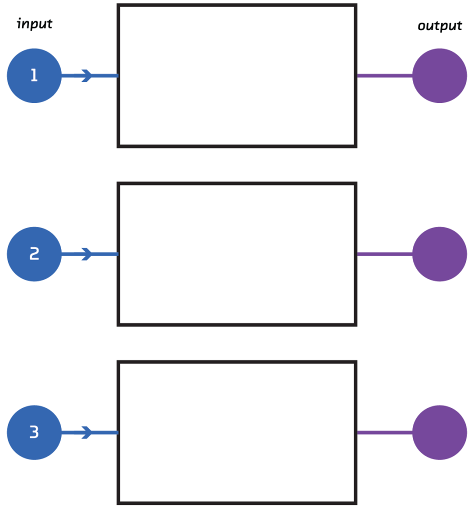
```python
def skipgram(words):
    for i in range(1, len(words)-1):
        yield words[i], (words[i-1], words[i+1])
```

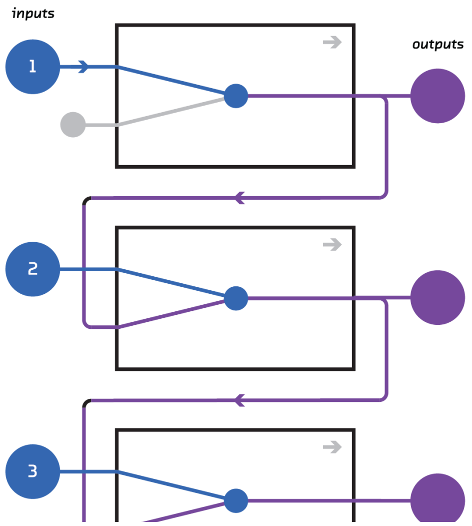


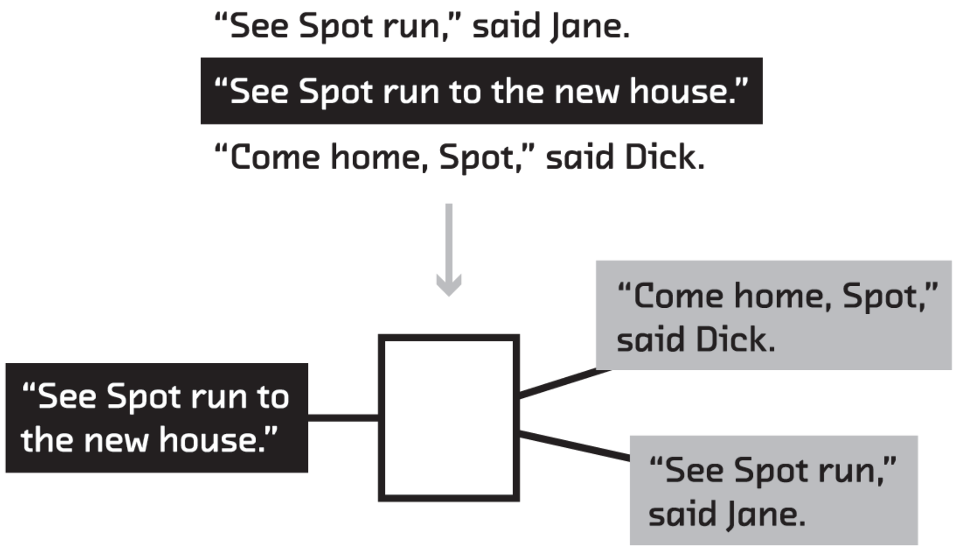
Berlin - Germany + France = Paris

```python
from keras.models import Model
from keras.layers import (Input, Embedding, Merge,
                          Lambda, Activation)

vector_size=300
word_index = Input(shape=1)
word_point = Input(shape=1)

syn0 = Embedding(len(vocab), vector_size)(word_index)
syn1= Embedding(len(vocab), vector_size)(word_point)

merge = Merge([syn0, syn1], mode='mul')
merge_sum = Lambda(lambda x: x.sum(axis=-1))(merge)
context = Activation('sigmoid')(merge_sum)

model = Model(input=[word, context], output=output)
model.compile(loss='binary_crossentropy', optimizer='adam')
```

Feed Forward vs Recurrent Network






“See Spot run,” said Jane.
“See Spot run to the new house.”
“Come home, Spot,” said Dick.
“See Spot run to the new house.”
“Come home, Spot,” said Dick.
“See Spot run,” said Jane.

RNN Summarization Sketch for Articles

1. Find articles summaries heavy on quotes (http://thebrowser.com/)
2. Score every sentence in the articles based on their "closeness" to a quote
3. Use `skip-thoughts` to encode every sentence in the article
4. Train and RNN to predict these scores given the sentence vector
5. Evaluate trained model on new things!

keras makes RNN's simple

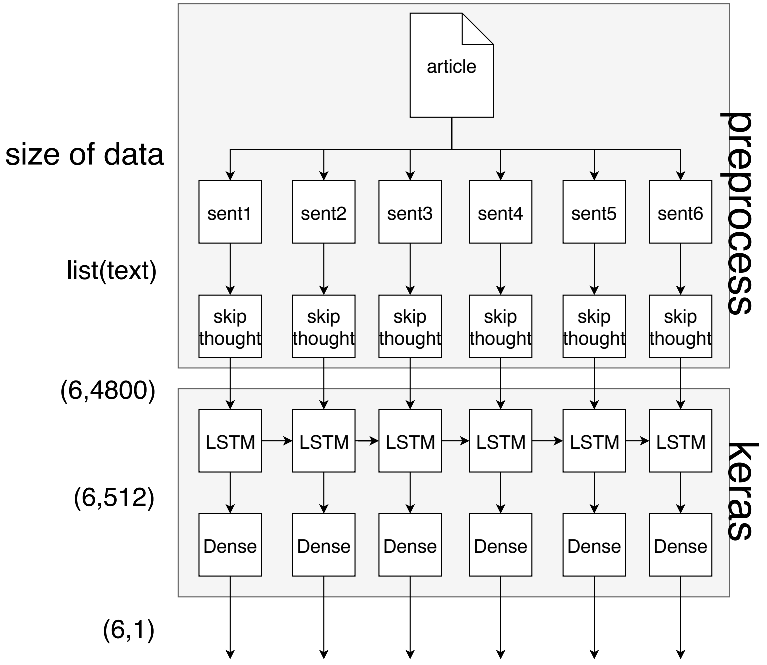
(http://keras.io/)

Example: proprocess

```
from skipthoughts import skipthoughts

from .utils import load_data

(articles, scores), (articles_test, scores_test) = load_data()
articles_vectors = skipthoughts.encode(articles)
articles_vectors_test = skipthoughts.encode(articles_test)
```

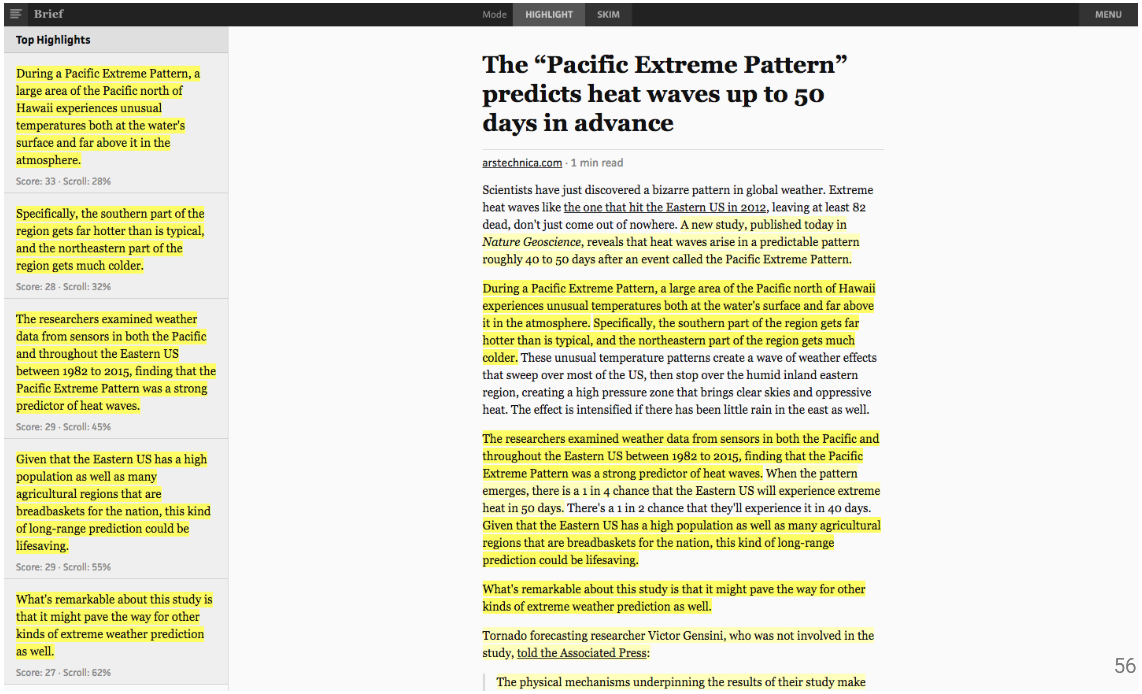
Example: model def and training

```
from keras.models import Model
from keras.layers.recurrent import LSTM
from keras.layers.core import Dense
from keras.layers.wrappers import TimeDistributed

model = Model()
model.add(LSTM(512, input_shape=(None, 4800),
               dropout_W=0.3, dropout_U=0.3))
model.add(TimeDistributed(Dense(1)))
model.compile(loss='mean_absolute_error', optimizer='rmsprop')

model.fit(articles_vectors, scores, validation_split=0.10)
loss, acc = model.evaluate(articles_vectors_test, scores_test)
print('Test loss / test accuracy = {:.4f} / {:.4f}'
      .format(loss, acc))

model.save("models/new_model.h5")
```


size of data
list(text)
(6,4800)
(6,512)
(6,1)
article
sent1
sent2
sent3
sent4
sent5
sent6
skip thought
LSTM
Dense
preprocess
keras

Example Model: evaluation

```python
from keras.models import load_model
from flask import Flask, request
import nltk

app = Flask(__name__)
model = load_model("models/new_model.h5")

@app.route('/api/evaluate', methods=['POST'])
def evaluate():
    article = request.data
    sentences = nltk.sent_tokenize(article)
    sentence_vectors = skipthoughts.encode(sentences)
    return model.predict(sentence_vectors)
```



Top Highlights

During a Pacific Extreme Pattern, a large area of the Pacific north of Hawaii experiences unusual temperatures both at the water's surface and far above it in the atmosphere.

Score: 33 · Scroll: 28%

Specifically, the southern part of the region gets far hotter than is typical, and the northeastern part of the region gets much colder.

Score: 28 · Scroll: 32%

The researchers examined weather data from sensors in both the Pacific and throughout the Eastern US between 1982 to 2015, finding that the Pacific Extreme Pattern was a strong predictor of heat waves.

Score: 29 · Scroll: 45%

Given that the Eastern US has a high population as well as many agricultural regions that are breadbaskets for the nation, this kind of long-range prediction could be lifesaving.

Score: 29 · Scroll: 55%

What's remarkable about this study is that it might pave the way for other kinds of extreme weather prediction as well.

Score: 27 · Scroll: 62%

The "Pacific Extreme Pattern" predicts heat waves up to 50 days in advance

arstechnica.com · 1 min read

Scientists have just discovered a bizarre pattern in global weather. Extreme heat waves like the one that hit the Eastern US in 2012, leaving at least 82 dead, don't just come out of nowhere. A new study, published today in *Nature Geoscience*, reveals that heat waves arise in a predictable pattern roughly 40 to 50 days after an event called the Pacific Extreme Pattern.

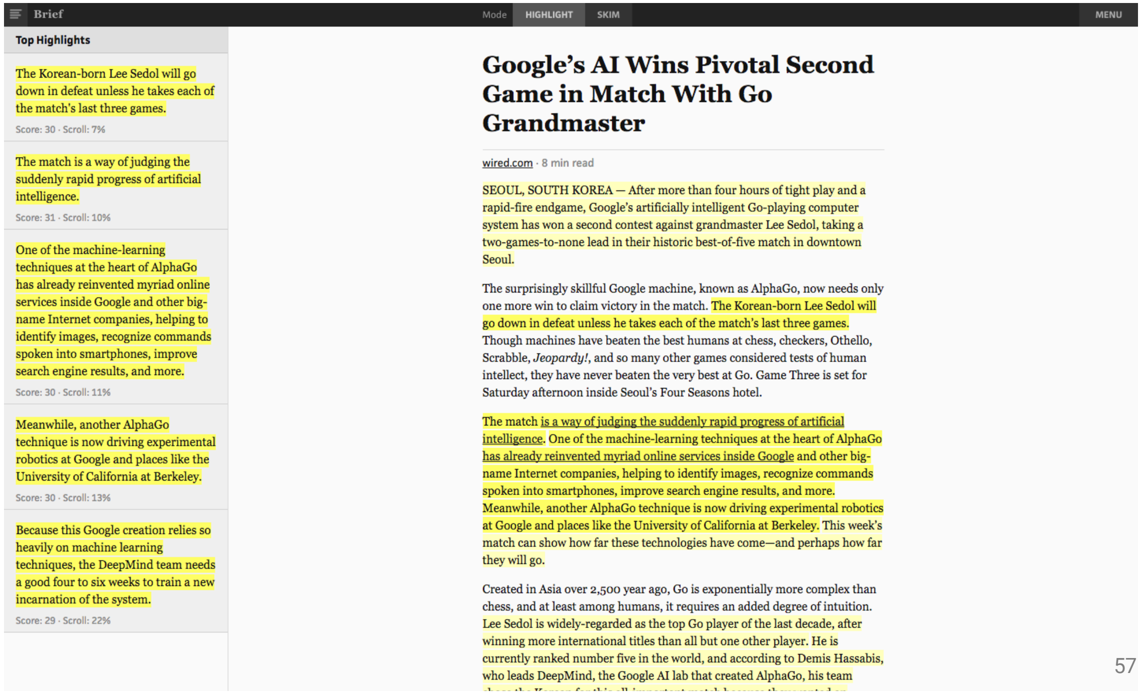
During a Pacific Extreme Pattern, a large area of the Pacific north of Hawaii experiences unusual temperatures both at the water's surface and far above it in the atmosphere. Specifically, the southern part of the region gets far hotter than is typical, and the northeastern part of the region gets much colder. These unusual temperature patterns create a wave of weather effects that sweep over most of the US, then stop over the humid inland eastern region, creating a high pressure zone that brings clear skies and oppressive heat. The effect is intensified if there has been little rain in the east as well.

The researchers examined weather data from sensors in both the Pacific and throughout the Eastern US between 1982 to 2015, finding that the Pacific Extreme Pattern was a strong predictor of heat waves. When the pattern emerges, there is a 1 in 4 chance that the Eastern US will experience extreme heat in 50 days. There's a 1 in 2 chance that they'll experience it in 40 days. Given that the Eastern US has a high population as well as many agricultural regions that are breadbaskets for the nation, this kind of long-range prediction could be lifesaving.

What's remarkable about this study is that it might pave the way for other kinds of extreme weather prediction as well.

Tornado forecasting researcher Victor Gensini, who was not involved in the study, told the Associated Press:

> The physical mechanisms underpinning the results of their study make



Top Highlights

The Korean-born Lee Sedol will go down in defeat unless he takes each of the match's last three games.

Score: 30 · Scroll: 7%

The match is a way of judging the suddenly rapid progress of artificial intelligence.

Score: 31 · Scroll: 10%

One of the machine-learning techniques at the heart of AlphaGo has already reinvented myriad online services inside Google and other big-name Internet companies, helping to identify images, recognize commands spoken into smartphones, improve search engine results, and more.

Score: 30 · Scroll: 11%

Meanwhile, another AlphaGo technique is now driving experimental robotics at Google and places like the University of California at Berkeley.

Score: 30 · Scroll: 13%

Because this Google creation relies so heavily on machine learning techniques, the DeepMind team needs a good four to six weeks to train a new incarnation of the system.

Score: 29 · Scroll: 22%

Google's AI Wins Pivotal Second Game in Match With Go Grandmaster

wired.com · 8 min read

SEOUL, SOUTH KOREA — After more than four hours of tight play and a rapid-fire endgame, Google's artificially intelligent Go-playing computer system has won a second contest against grandmaster Lee Sedol, taking a two-games-to-none lead in their historic best-of-five match in downtown Seoul.

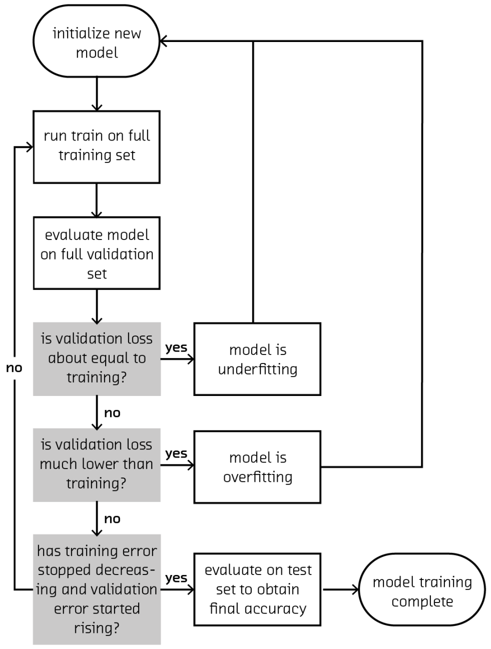
The surprisingly skillful Google machine, known as AlphaGo, now needs only one more win to claim victory in the match. The Korean-born Lee Sedol will go down in defeat unless he takes each of the match's last three games. Though machines have beaten the best humans at chess, checkers, Othello, Scrabble, *Jeopardy!*, and so many other games considered tests of human intellect, they have never beaten the very best at Go. Game Three is set for Saturday afternoon inside Seoul's Four Seasons hotel.

The match is a way of judging the suddenly rapid progress of artificial intelligence. One of the machine-learning techniques at the heart of AlphaGo has already reinvented myriad online services inside Google and other big-name Internet companies, helping to identify images, recognize commands spoken into smartphones, improve search engine results, and more. Meanwhile, another AlphaGo technique is now driving experimental robotics at Google and places like the University of California at Berkeley. This week's match can show how far these technologies have come—and perhaps how far they will go.

Created in Asia over 2,500 year ago, Go is exponentially more complex than chess, and at least among humans, it requires an added degree of intuition. Lee Sedol is widely-regarded as the top Go player of the last decade, after winning more international titles than all but one other player. He is currently ranked number five in the world, and according to Demis Hassabis, who leads DeepMind, the Google AI lab that created AlphaGo, his team

Thoughts of doing this method

- Scoring function used is SUPER important
- Hope you have a GPU
- Hyper-parameters for all!
- Structure of model can change where it's applicable
- SGD means random initialization... may need to fit multiple times


initialize new model
run train on full training set
evaluate model on full validation set
is validation loss about equal to training?
yes
model is underfitting
no
is validation loss much lower than training?
yes
model is overfitting
no
has training error stopped decreas-ing and validation error started rising?
yes
evaluate on test set to obtain final accuracy
model training complete
no

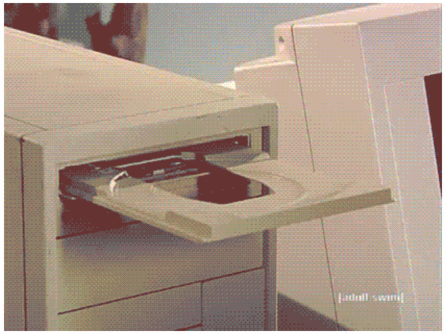
REGULARIZE!

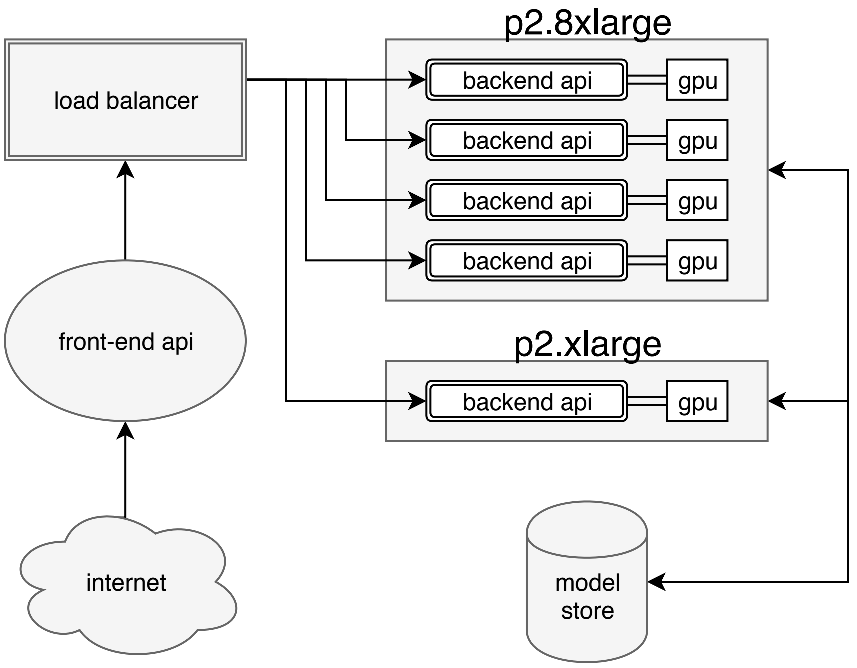
- **dropout**: only parts of the NN participate in every round
- **l1/l2**: add penalty term for large weights
- **batchnormalization**: unit mean/std for each batch of data

VALIDATE AND TEST!

lots of parameters
==
potential for overfitting

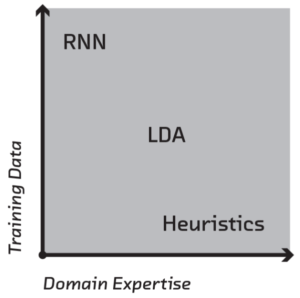
deploy?

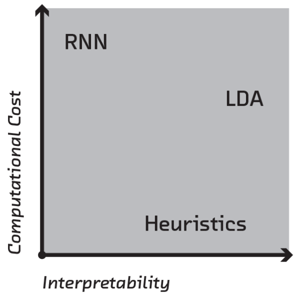


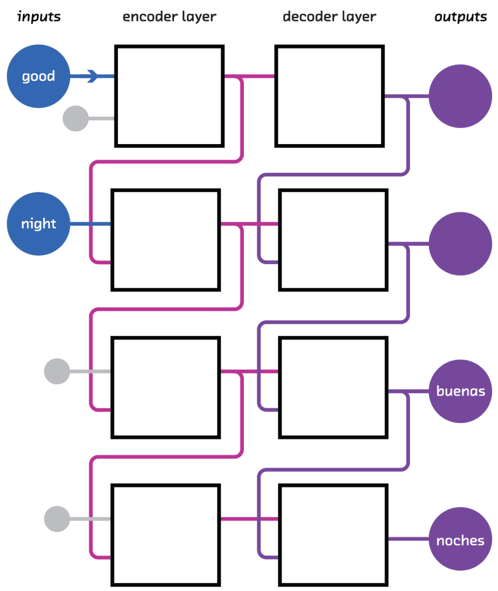

load balancer
front-end api
internet
p2.8xlarge
backend api
gpu
backend api
gpu
backend api
gpu
backend api
gpu
p2.xlarge
backend api
gpu
model store

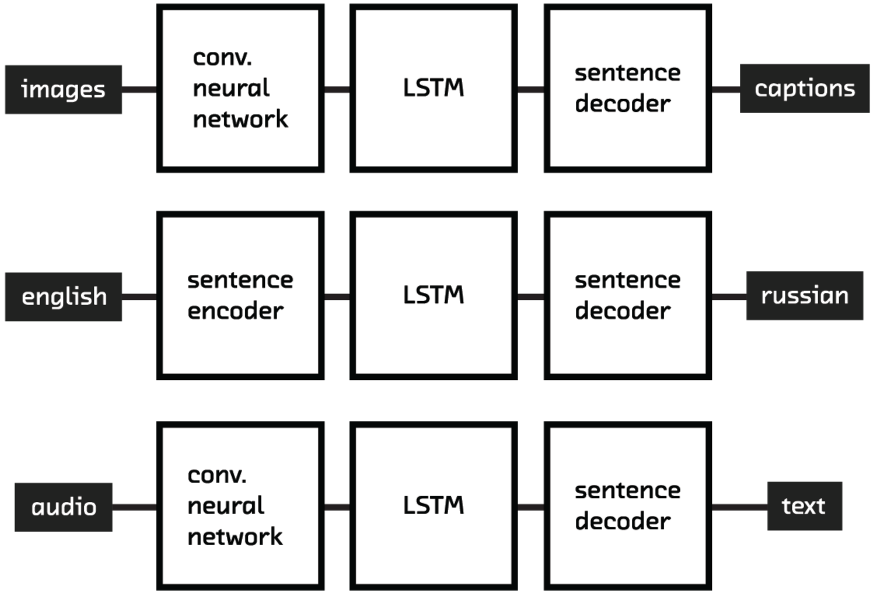
careful: data drift

set monitoring on the distribution of results


RNN
LDA
Heuristics
Training Data
Domain Expertise


RNN
LDA
Heuristics
Computational Cost
Interpretability


inputs
encoder layer
decoder layer
outputs
good
night
buenas
noches


images
conv. neural network
LSTM
sentence decoder
captions
english
sentence encoder
LSTM
sentence decoder
russian
audio
conv. neural network
LSTM
sentence decoder
text

Emerging Applications

- Text Generation
- Audio Generation
- Event Detection
- Intent Identification
- Decision Making

Fast Forward Labs