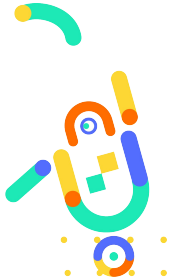
GDG Hanoi

Google I/O 18 Extended Hanoi

Convolutional Neural Network in Tensorflow/Keras for Natural Language Processing and Digital Image Processing

Present by Nguyen Duc Tam
CTO at AimeSoft

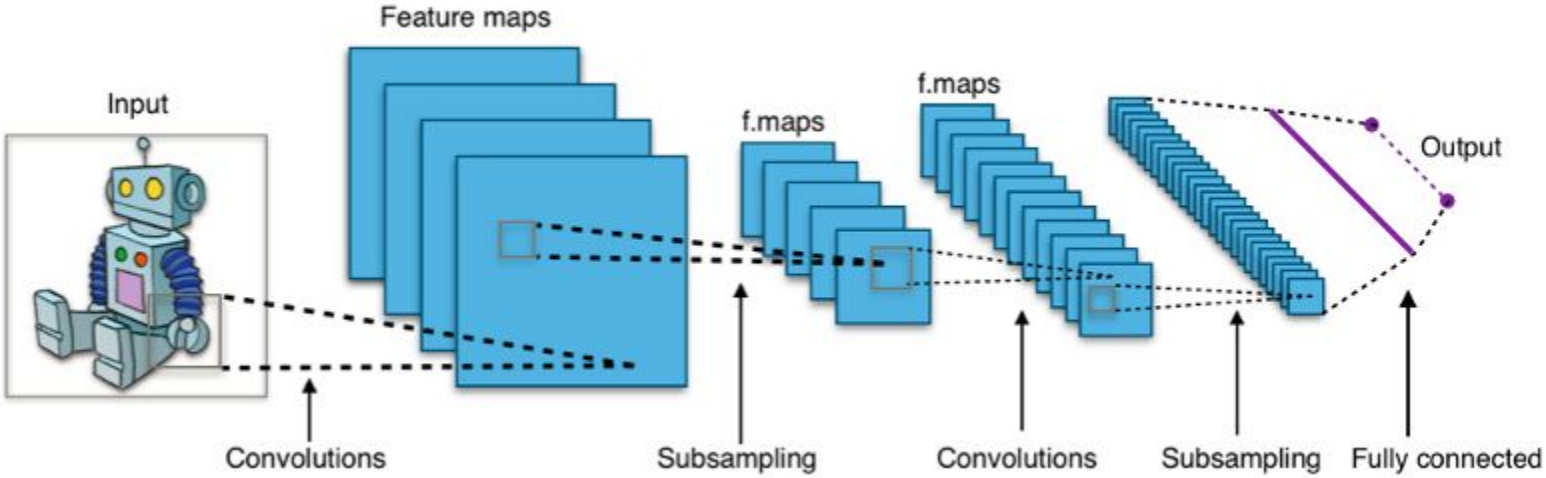


Who is this guy?

- received PhD in Mathematics (Singularity Theory) from the University of Tokyo after 2 years and 4 months
- worked at Anduin Transactions - a platform for private market transactions, mainly focused on Legal Document Parsing and Generation
- currently work at Aimesoft, supervise several Image Recognition and Natural Language Processing projects

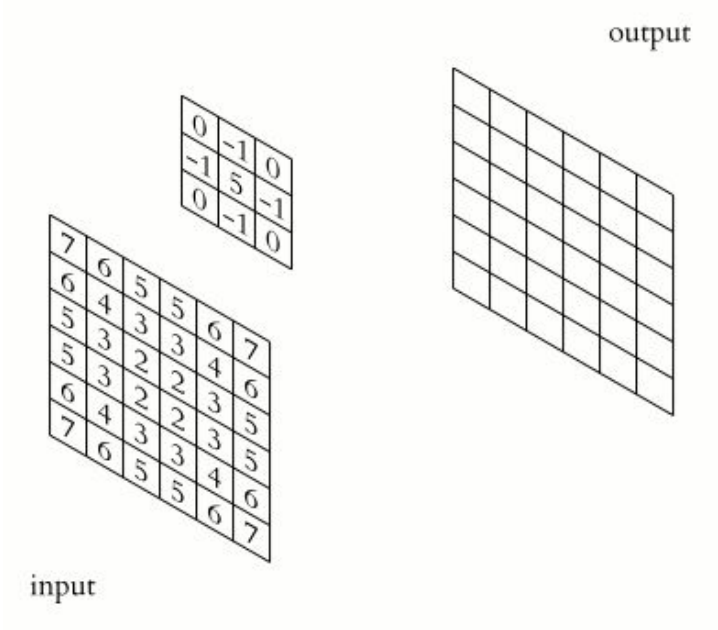
Contact me at @tamnd87 or tamnd@aimesoft.com

What's Convolutional Neural Network (CNN)?



Typical CNN architecture

Convolution



1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

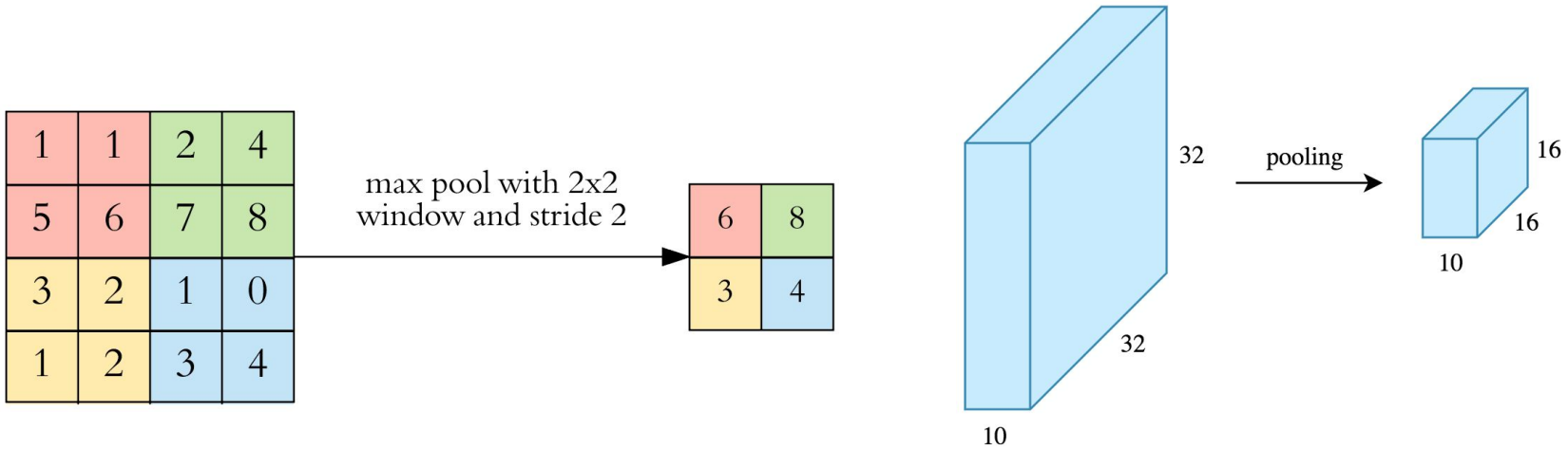
Input x Filter

4		

Feature

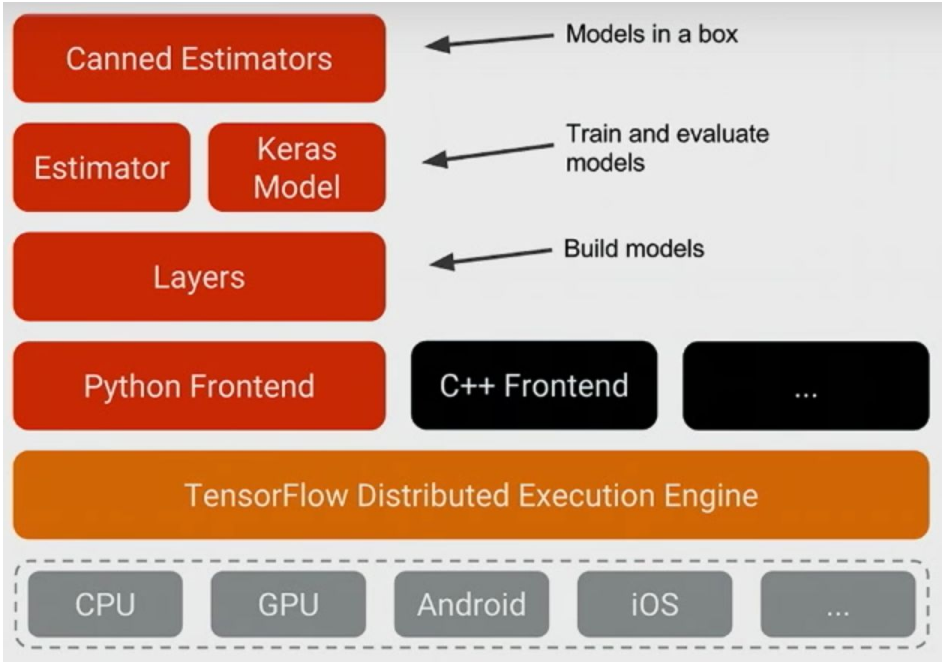
[3D Convolution Animation](#)—Author: Michael Plotke

Pooling



Pooling example — Author: Arden Dertat

Tensorflow



Source: Google Blog

κέρας overview

What is Keras?

- Neural Network library written in python
- Design to be simple and straightforward
- Built on top of different deep learning libraries such as Tensorflow, Theano and CNTK

Why Keras?

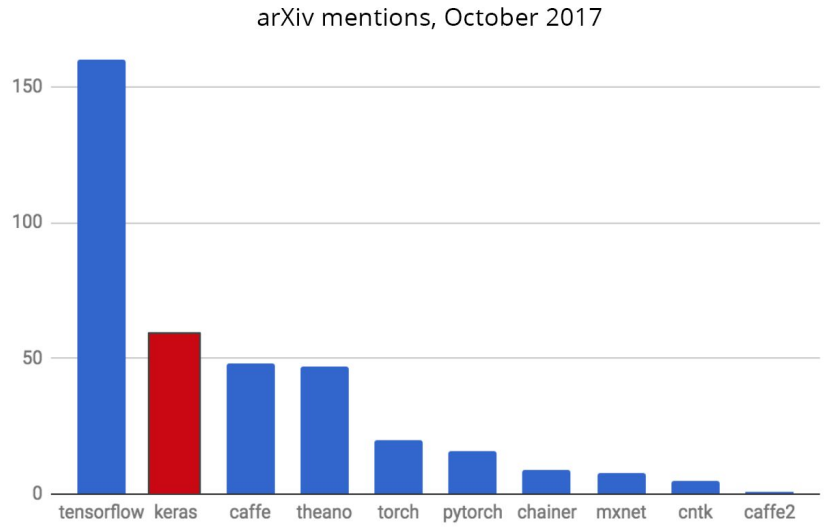
- Simple
- Highly modular
- Deep enough to build models

Source: Aly Osama

Why use Keras?

- Keras prioritizes developers experience
- Keras has broad adoption
- Keras makes it easy to turn models into product
- Keras supports multiple backend engines
- Keras has strong multi-GPU support and distributed training support
- Keras development is backed by key companies in the deep learning ecosystem

Google Microsoft NVIDIA aws



Source: keras.io

When to use Keras?

If you're a beginner and interested in quickly implementing your ideas

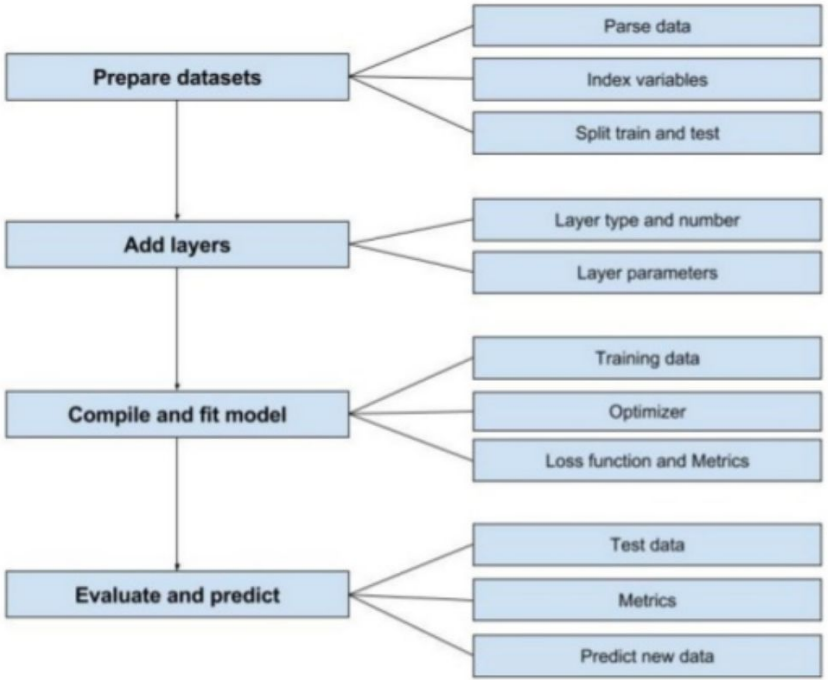
- Python + **Keras**: Super fast implementation, good extensibility

If you want to do fundamental research in Deep Learning

- Python + **Tensorflow** or PyTorch: Excellent extensibility

Source: [Deep learning using Keras]

Keras in 30 seconds (1)



1. Sequential model (a linear stack of layers)

```
from keras.models import Sequential
model = Sequential()
```

```
from keras.layers import Dense, Activation

model.add(Dense(units=64, input_dim=100))
model.add(Activation('relu'))
model.add(Dense(units=10))
model.add(Activation('softmax'))
```


Source: [Deep learning using Keras]

Keras in 30 seconds (2)

2. Compile Model

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

Also, you can further configure your optimizer

```
model.compile(loss=keras.losses.categorical_crossentropy,
optimizer=keras.optimizers.SGD(lr=0.01, momentum=0.9))
```

3. Training

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

You can feed data batches manualy

```
model.train_on_batch(x_batch, y_batch)
```

4. Evaluation

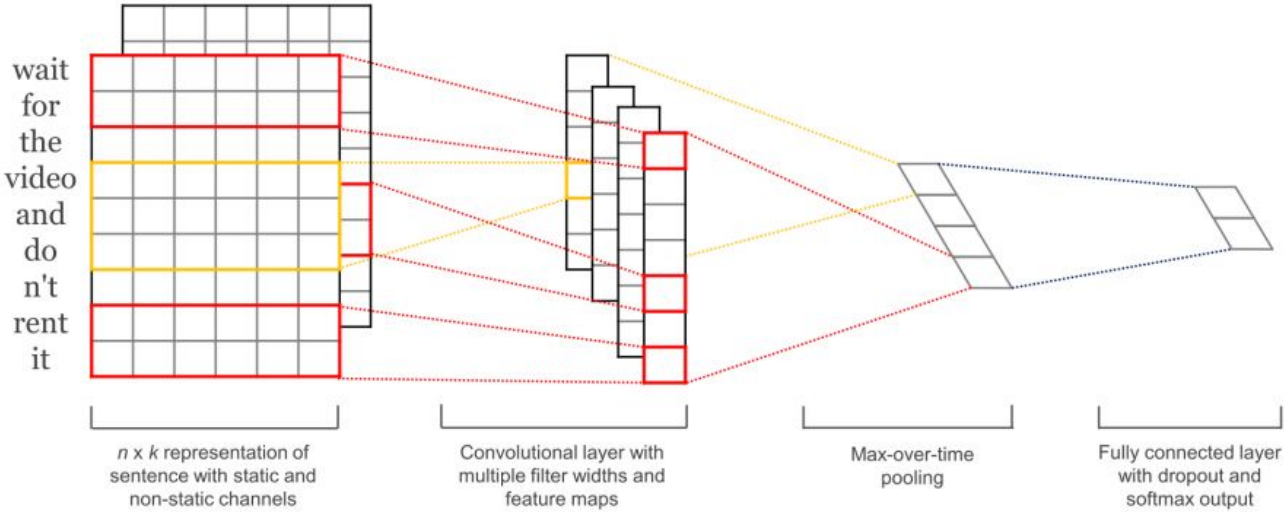
```
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

5. Prediction

```
classes = model.predict(x_test, batch_size=128)
```


Source: [Deep learning using Keras]

CNN for Natural Language Processing

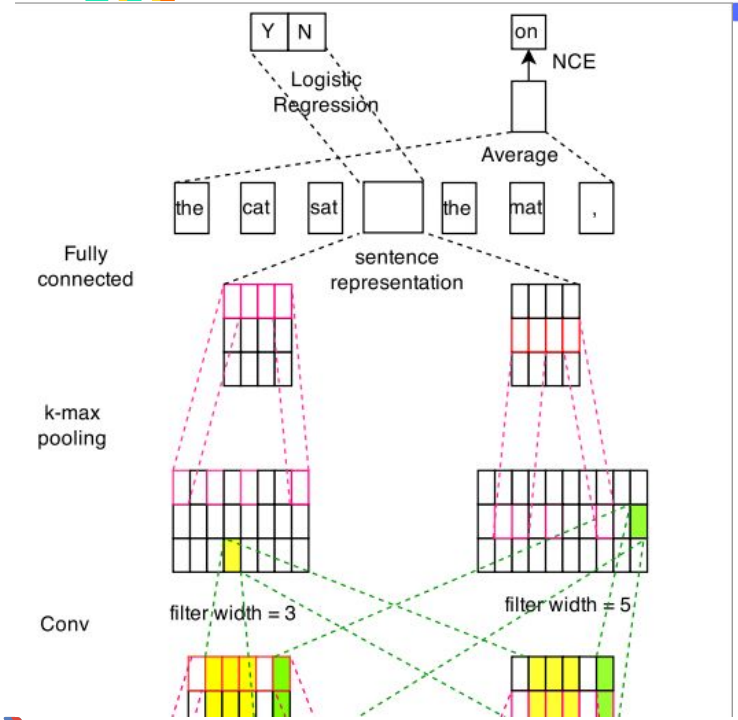


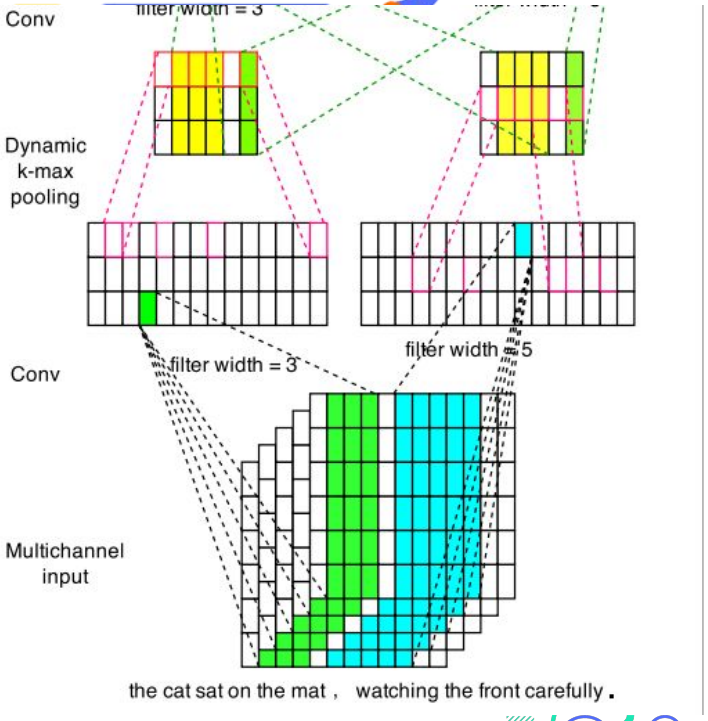
Convolutional Neural Network for Sentence Classification - Yoon Kim (2014)

Applications of CNN in NLP

- Text classification and categorization
- Named Entity Recognition (NER)
- Part-of-speech Tagging
- Semantic Parsing and Question Answering
- Paraphrase Detection
- Language Generation and Multi-document Summarization
- Machine Translation
- Speech Recognition
- Character Recognition
- Spell Checking

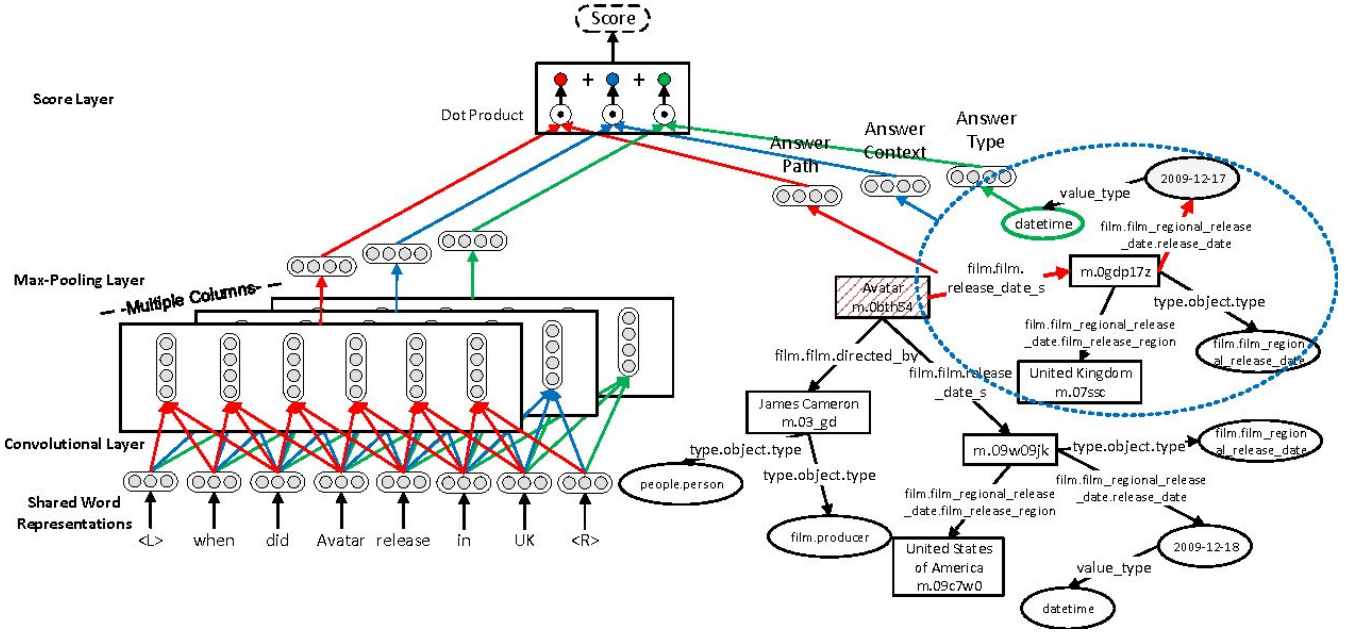
Multichannel Variable-Size Convolution





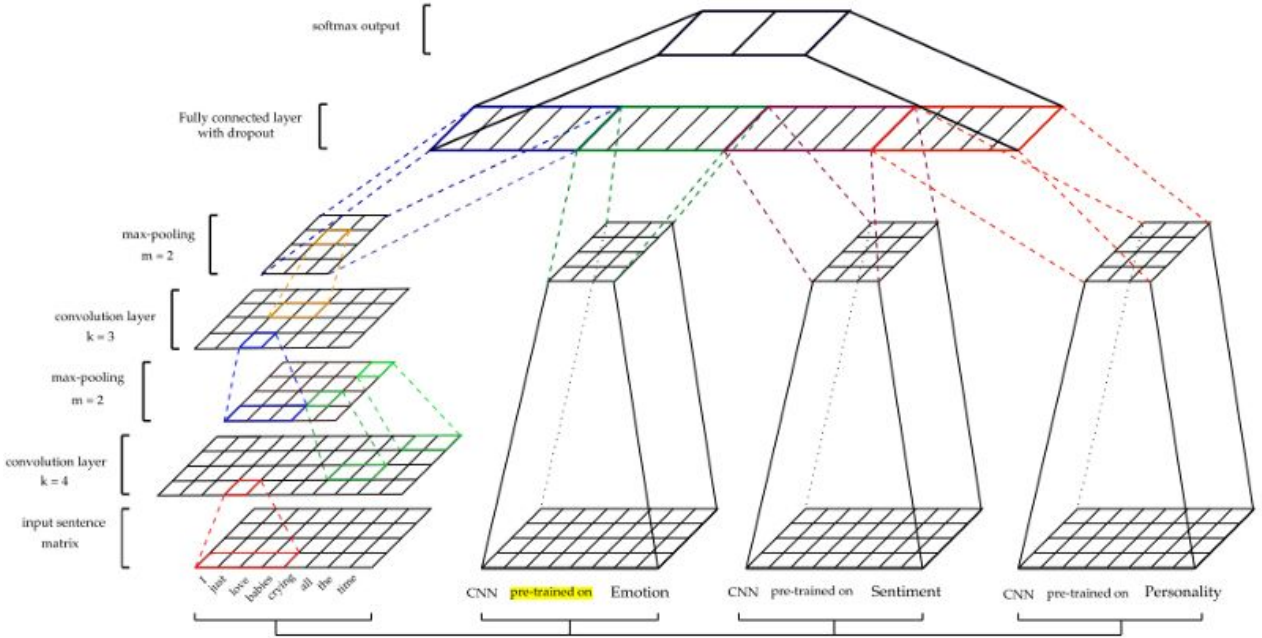
Author: Wenpeng Yin, Hinrich Schütze

Question answering with multi-column CNNs



Author: Li Dong, Furu Wei

Detecting Sarcasm with CNNs



Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Prateek Vij (NTU - Singapore)

GDG Hanoi

Google I/O 18 Extended Hanoi

Named Entity Recognition (NER) API

Input
"Hôm nay tôi đến Trống Đồng Palace để tham dự Google IO Hà Nội."

- CNN for character-level embeddings
- LSTM (Long Short-Term Memory network) for sequence labelling
- CRF (Conditional Random Field) for entity classification

State-of-the-art performance for NER

output.json

```
[
  {
    "entity": "DATE",
    "surface": "Hôm nay",
    "value": "2018-07-21"
  },
  {
    "entity": "ORG",
    "surface": "Google IO Hà Nội",
    "value": "Google IO Hà Nội"
  },
  {
    "entity": "PROVINCE",
    "surface": "Hà Nội",
    "value": {
      "province": "Hà Nội"
    }
  },
  {
    "entity": "loc",
    "surface": "Trống Đồng Palace",
    "value": "Trống Đồng Palace"
  }
]
```

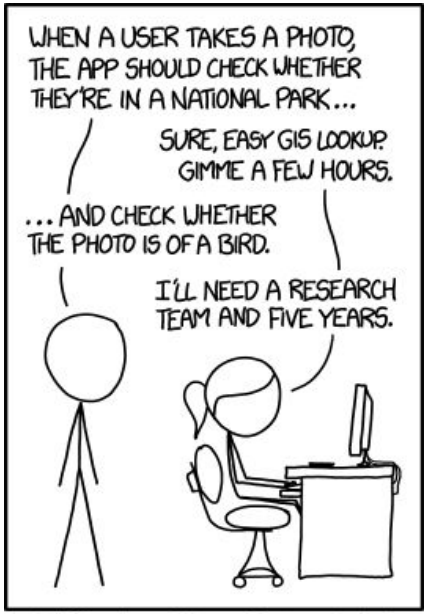
* 390 output.json JSON unix | 24: 0 All

GDG Hanoi

Nguyễn Tuấn Đức, Phạm Thế Quyền
(NLP Engineer @ Aimesoft)

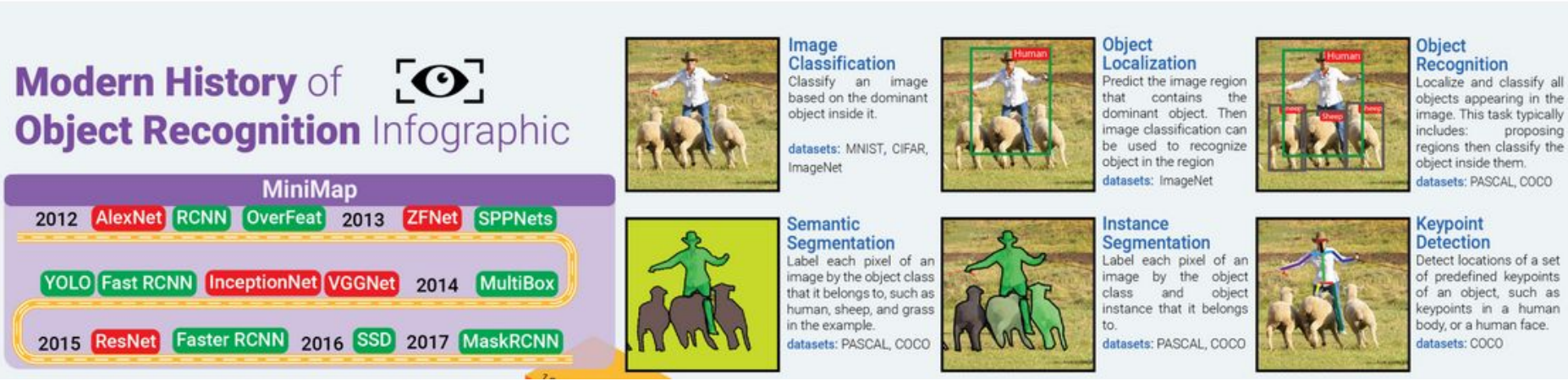
Google I/O 18 Extended Hanoi

CNN for Computer Vision



Source: xkcd.com/1425

The modern History of Object Recognition

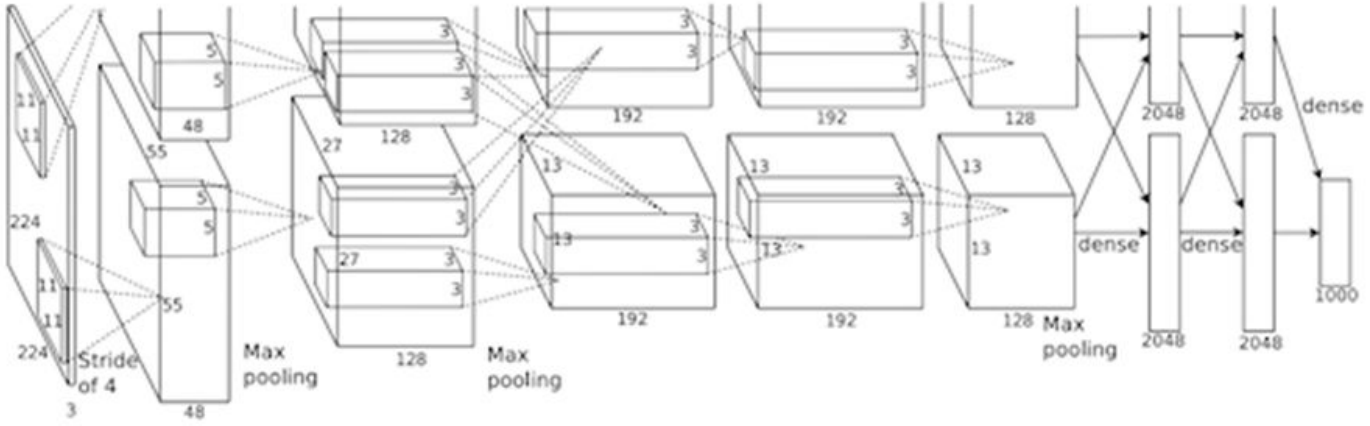


Source: Medium - Author: Dang Ha The Hien (PhD at UiO)

Applications of CNN in Computer Vision

- Image Classification
- Image Retrieval
- Object Detection
- Semantic Segmentation
- Similarity Learning
- Image Captioning
- Generative models
- Video Classification

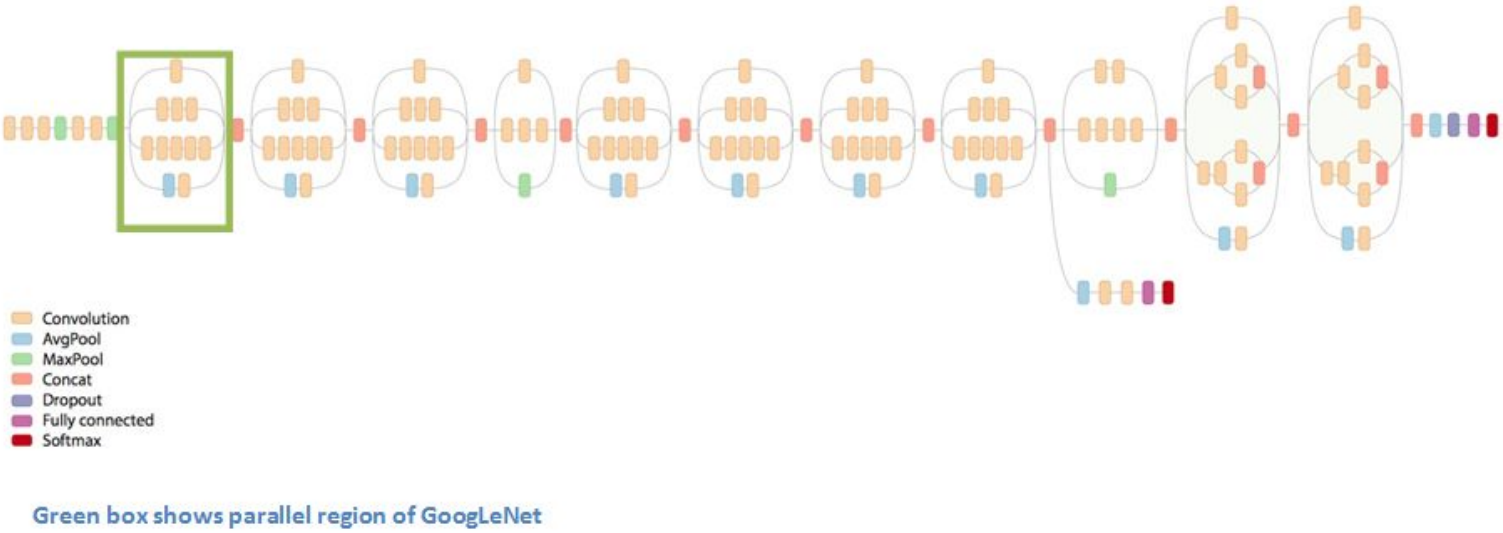
AlexNet (2012)



AlexNet architecture (May look weird because there are two different "streams". This is because the training process was so computationally expensive that they had to split the training onto 2 GPUs)

Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton

GoogLeNet (2015)



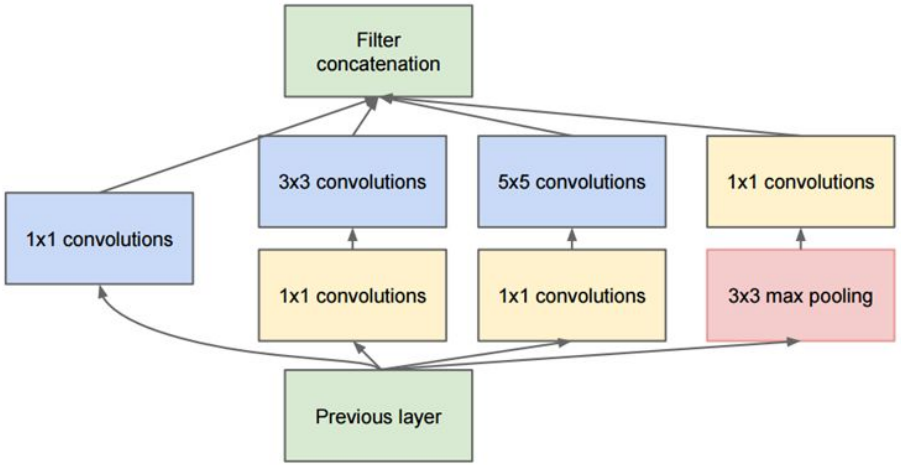
Green box shows parallel region of GoogLeNet

Going Deeper with Convolutions (Google)

Going deeper...

- Used 9 Inception module, with over 100 layers
- Use an average pool, to go from 7x7x1024 volume to 1x1x1024 volume
- Trained on “a few high-end GPUs within a week”

Really set the stage for some amazing architectures that we could see in the coming years.

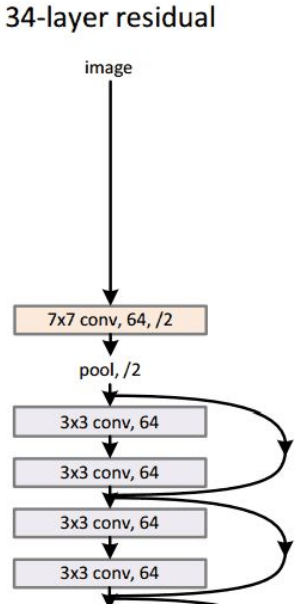


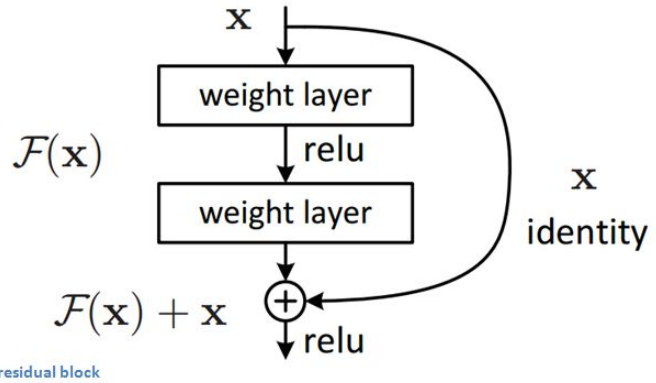
Full Inception module

Well said Leo, well said

Well said Leo, well said

Microsoft ResNet (2015)



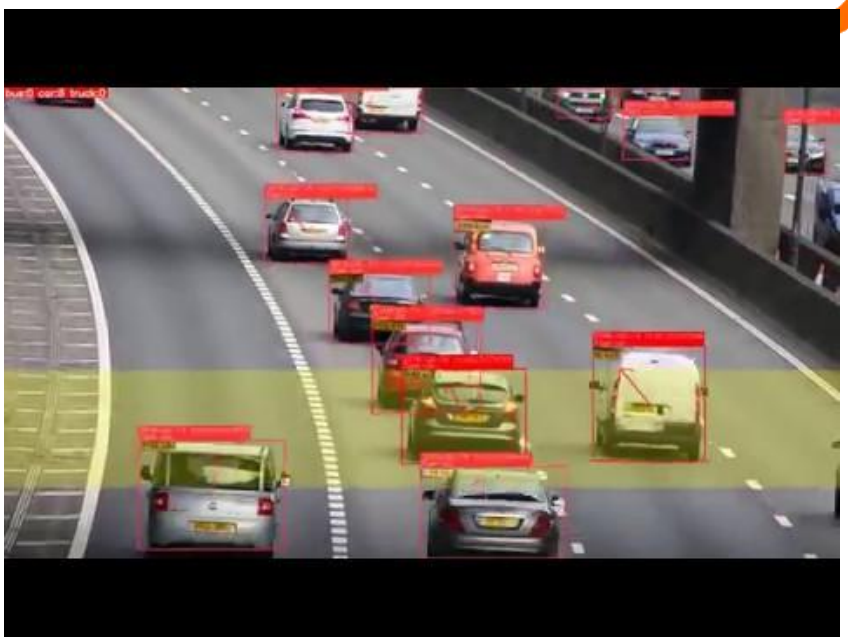


A residual block

- “Ultra-deep” - Yann LeCun
- 152 layers…
- Trained on an 8 GPU machine for two to three weeks

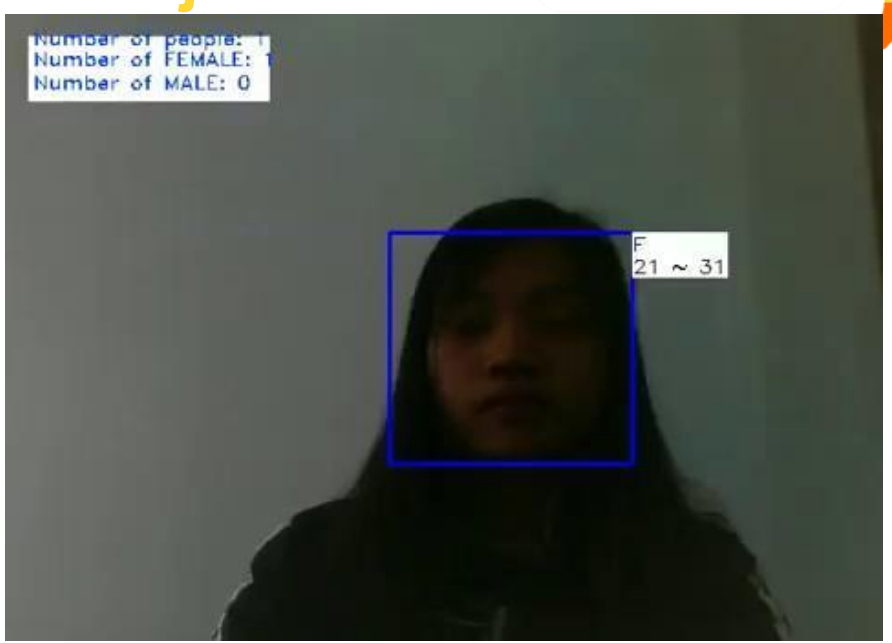
Demo time

Traffic monitor



Nguyễn Tất Nguyên (Computer Vision Engineer @ Aimesoft)

Real-time object detection



Nguyễn Quang Bình, Nguyễn Thị Vân Anh
(Computer Vision Engineer @ Aimesoft)

Conclusion

- CNNs are widely used in various applications in NLP and Computer Vision
- It is easy to write CNNs code on TF/Keras
- We successfully deployed CNNs on TF/Keras for many applications in banking and transportation which require high performance and high accuracy.

Thank you!

@tamnd87

tamnd@aimesoft.com