Applications and Deep Learning State of the Art
Image Recognition

- Imagenet is the standard benchmark set for image recognition
- Classify 256x256 images into 1000 categories, such as "person", "bike", "cheetah", etc.
- Total 1.2M images
- Many error metrics, including top-5 error: error rate with 5 guesses

Picture from Alex Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks", 2012
Computer Vision: Case Visy Oy

- Computer vision for logistics since 1994
- License plates (LPR), container codes,…
- How to grow in an environment with heavy competition?
  - Be agile
  - Be innovative
  - Be credible
  - Be customer oriented
  - Be technologically state-of-the-art
What has changes in 20 years?

• In 1996:
  – Small images (e.g., 10x10)
  – Few classes (< 100)
  – Small network (< 4 layers)
  – Small data (< 50K images)

• In 2016:
  – Large images (256x256)
  – Many classes (> 1K)
  – Deep net (> 100 kerrosta)
  – Large data (> 1M)
Net Depth Evolution Since 2012

ILSVRC Image Recognition Task:

- 1.2 million images
- 1,000 categories

(Prior to 2012: 25.7 %)

<table>
<thead>
<tr>
<th>Team</th>
<th>Year</th>
<th>Place</th>
<th>Error (top-5)</th>
<th>Uses external data</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuperVision</td>
<td>2012</td>
<td>1st</td>
<td>16.4%</td>
<td>no</td>
</tr>
<tr>
<td>SuperVision</td>
<td>2014</td>
<td>1st</td>
<td>15.3%</td>
<td>Imagenet 22k</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>1st</td>
<td>11.7%</td>
<td>no</td>
</tr>
<tr>
<td>Clarifai</td>
<td>2013</td>
<td>1st</td>
<td>11.2%</td>
<td>Imagenet 22k</td>
</tr>
<tr>
<td>MSRA</td>
<td>2014</td>
<td>3rd</td>
<td>7.35%</td>
<td>no</td>
</tr>
<tr>
<td>VGG</td>
<td>2014</td>
<td>2nd</td>
<td>7.32%</td>
<td>no</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>2014</td>
<td>1st</td>
<td>6.67%</td>
<td>no</td>
</tr>
</tbody>
</table>

- 2015 winner: MSRA (error 3.57%)
- 2016 winner: Trimps-Soushen (2.99 %)
- 2017 winner: Uni Oxford (2.25 %)

8 layers
16 layers
22 layers
152 layers
152 layers (but many nets)
101 layers (many nets, layers were blocks)
ILSVRC2012

• ILSVRC2012\(^1\) was a game changer
• ConvNets dropped the top-5 error 26.2\% → 15.3 \%.
• The network is now called *AlexNet* named after the first author (see previous slide).
• Network contains 8 layers (5 convolutional followed by 3 dense); altogether 60M parameters.

\(^1\) Imagenet Large Scale Visual Recognition Challenge
The AlexNet

- The architecture is illustrated in the figure.
- The pipeline is divided to two paths (upper & lower) to fit to 3GB of GPU memory available at the time (running on 2 GPU’s)
- Introduced many tricks for *data augmentation*
- Left-right flip
- Crop subimages (224x224)

Picture from Alex Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks", 2012
Since 2012, ConvNets have dominated. In 2014 there were 2 almost equal teams:
- GoogLeNet Team with 6.66% Top-5 error
- VGG Team with 7.33% Top-5 error
In some subchallenges VGG was the winner.
GoogLeNet: 22 layers, only 7M parameters due to fully convolutional structure and clever \textit{inception} architecture.
VGG: 16 layers, 144M parameters.
ILSVRC2015

- Winner MSRA (Microsoft Research) with TOP-5 error 3.57 %
- 152 layers! 51M parameters.
- Built from residual blocks (which include the inception trick from previous year)
- Key idea is to add identity shortcuts, which make training easier
Some Famous Networks

Pretraining

- With small data, people often initialize the net with a *pretrained* network.
- This may be one of the imagenet winners; VGG16, ResNet, …
- See `keras.applications` for some of these.

VGG16 network
source: [https://www.cs.toronto.edu/~frossard/post/vgg16/](https://www.cs.toronto.edu/~frossard/post/vgg16/)
Example: Cats vs. Dogs

• Let’s study the effect of pretraining with classical image recognition task: learn to classify images to cats and dogs.
• We use the Oxford Cats and Dogs dataset.
• Subset of 3687 images of the full dataset (1189 cats; 2498 dogs) for which the ground truth location of the animal’s head is available.
Network 1: Design and Train from Scratch

```python
# Initialize the model
model = Sequential()

shape = (64, 64, 3)

# Add six convolutional layers. Maxpool after every second convolution.
model.add(Conv2D(filters=32, kernel_size=3, padding="same", activation="relu", input_shape=shape))
model.add(Conv2D(filters=32, kernel_size=3, padding="same", activation="relu"))
model.add(MaxPooling2D(2, 2))  # Shrink feature maps to 32x32

model.add(Conv2D(filters=48, kernel_size=3, padding="same", activation="relu"))
model.add(Conv2D(filters=48, kernel_size=3, padding="same", activation="relu"))
model.add(MaxPooling2D(2, 2))  # Shrink feature maps to 16x16

model.add(Conv2D(filters=64, kernel_size=3, padding="same", activation="relu"))
model.add(Conv2D(filters=64, kernel_size=3, padding="same", activation="relu"))
model.add(MaxPooling2D(2, 2))  # Shrink feature maps to 8x8

# Vectorize the 8x8x64 representation to 4096x1 vector
model.add(Flatten())

# Add a dense layer with 128 nodes
model.add(Dense(128, activation="relu"))
model.add(Dropout(0.5))

# Finally, the output layer has 1 output with logistic sigmoid nonlinearity
model.add(Dense(1, activation="sigmoid"))
```
Network 1: Design and Train from Scratch
Network 2: Start from a Pretrained Network

```
# Import the network container and the three types of layers
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.layers import Dense

# Initialize the VGG16 network. Omit the dense layers on top.
base_model = VGG16(include_top = False, weights = "imagenet",
 input_shape = (64, 64, 3))

# We use the functional API, and grab the VGG16 output here:
w = base_model.output

# Now we can perform operations on w. First flatten it to 4096-dim vector:
w = Flatten()(w)

# Add dense layer:
w = Dense(128, activation = "relu")(w)

# Add output layer:
output = Dense(1, activation = "sigmoid")(w)

# Prepare the full model from input to output:
model = Model(inputs = [base_model.input], outputs = [output])

# Also set the Last Conv block (3 layers) as trainable.
# There are four layers above this block, so our indices
# start at -5 (i.e., last minus five):
model.layers[-5].trainable = True
model.layers[-6].trainable = True
model.layers[-7].trainable = True
```

VGG16 network source: [https://www.cs.toronto.edu/~frossard/post/vgg16/](https://www.cs.toronto.edu/~frossard/post/vgg16/)
Results

Trained from scratch

Initialized with VGG
Research at TUT

Images

Recognized as [6 8 3 7 4 6 5 1]. True #: [6 8 3 7 4 6 5 1].

Sound

Neural Net

Positive

Negative

Neutral

Text

Heikki Huttunen @heikbutt... 13.4.2016
Viime vuonna ylätyn erittäin positiivisesti. Suosittelem!

Rakkauden Wappuradio @wap...
Woop woop alle viikko lähteenä akkuna! Käykää tsekkaamassa ohjelmat wappuradio.fi niin tiedätte mitä kuunnella!#wappuradio #hype

66.0%
64.0%
62.0%
60.0%
58.0%
56.0%
54.0%

Traditional (GMM-HMM)
Multilabel DNN
Multilabel LSTM
Example case

- TUT has studied shallow convolutional architectures for fast/real time detection tasks
- For example, Automatic Car Type Detection from Picture: Van, Bus, Truck or Normal Vehicle
- The network recognizes the car type (4 classes) with 98% accuracy (13 000 images).
Components of the Network

- **Convolution**: 5x5 window
- **Maxpooling**: 2x2 downsampling with the maximum
- **Relu**: $\max(x, 0)$
- **Matrix multiplication**
- **Softmax**: 
  $$\sigma(x)_k = \frac{\exp(x_k)}{\sum \exp(x_k)}$$
Recurrent Networks

- Recurrent networks process sequences of arbitrary length; e.g.,
  - Sequence → sequence
  - Image → sequence
  - Sequence → class ID

Picture from [http://karpathy.github.io/2015/05/21/rnn-effectiveness/](http://karpathy.github.io/2015/05/21/rnn-effectiveness/)
Recurrent Networks

- Recurrent net consist of special nodes that remember past states.
- Each node receives 2 inputs: the data and the previous state.
- Keras implements SimpleRNN, LSTM and GRU layers.
- Most popular recurrent node type is Long Short Term Memory (LSTM) node.
- LSTM includes also gates, which can turn on/off the history and a few additional inputs.

Recurrent Networks

- An example of use is from our recent paper.
- We detect acoustic events within 61 categories.
- LSTM is particularly effective because it remembers the past events (or the context).
- In this case we used a bidirectional LSTM, which remembers also the future.
- BLSTM gives slight improvement over LSTM.
LSTM in Keras

- LSTM layers can be added to the model like any other layer type.
- This is an example for natural language modeling: *Can the network predict next symbol from the previous ones?*
- Accuracy is greatly improved from N-Gram etc.

```python
model = Sequential()
model.add(LSTM(512, return_sequences=True, input_shape=(maxlen, len(symbols))))
model.add(Dropout(0.2))
model.add(LSTM(512, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(len(symbols)))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='rmsprop')
```
Text Modeling

- The input to LSTM should be a sequence of vectors.
- For text modeling, we represent the symbols as binary vectors.

```python
from sklearn import preprocessing

lb = preprocessing.LabelBinarizer()
symbol_list = list("hello world")
lb.fit(symbol_list)
binary_table = lb.transform(symbol_list)
```
Text Modeling

- The prediction target for the LSTM net is simply the input delayed by one step.
- For example: we have shown the net these symbols: ['h', 'e', 'l', 'l', 'o', '_', 'w']
- Then the network should predict 'o'.

```
H [0, 0, 0, 1, 0, 0, 0, 0]  LSTM  [0, 0, 1, 0, 0, 0, 0, 0]  E
E [0, 0, 1, 0, 0, 0, 0, 0]  LSTM  [0, 0, 0, 1, 0, 0, 0, 0]  L
L [0, 0, 0, 0, 1, 0, 0, 0]  LSTM  [0, 0, 0, 0, 1, 0, 0, 0]  L
L [0, 0, 0, 0, 0, 1, 0, 0]  LSTM  [0, 0, 0, 0, 0, 1, 0, 0]  O
O [0, 0, 0, 0, 0, 0, 1, 0]  LSTM  [1, 0, 0, 0, 0, 0, 0, 0]  _
_ [1, 0, 0, 0, 0, 0, 0, 0]  LSTM  [0, 0, 0, 0, 0, 0, 1, 0]  W
W [0, 0, 0, 0, 0, 0, 0, 1]  LSTM  [0, 0, 0, 0, 0, 1, 0, 0]  O
```
Text Modeling

- Trained LSTM can be used as a text generator.
- Show the first character, and set the predicted symbol as the next input.
- Randomize among the top scoring symbols to avoid static loops.
Many LSTM Layers

- A straightforward extension of LSTM is to use it in multiple layers (typically less than 5).
- Below is an example of two layered LSTM.
- Note: Each blue block is exactly the same with, e.g., 512 LSTM nodes. So is each red block.

\[
\begin{align*}
[0, 0, 0, 1, 0, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 1, 0, 0, 0, 0, 0] \\
[0, 0, 1, 0, 0, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 0, 1, 0, 0, 0, 0] \\
[0, 0, 0, 0, 1, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 0, 0, 1, 0, 0, 0] \\
[0, 0, 0, 0, 1, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 0, 0, 1, 0, 0, 0] \\
[1, 0, 0, 0, 0, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 0, 0, 1, 0, 0, 0] \\
[0, 0, 0, 0, 1, 0, 0, 0] & \xrightarrow{\text{LSTM}} [0, 0, 0, 0, 1, 0, 0, 0] \\
\end{align*}
\]
LSTM Training

- LSTM net can be viewed as a very deep non-recurrent network.
- The LSTM net can be *unfolded* in time over a sequence of time steps.
- After unfolding, the normal gradient based learning rules apply.

---

Text Modeling Experiment

• Keras includes an example script: https://github.com/fchollet/keras/blob/master/examples/lstm_text_generation.py

• Train a 2-layer LSTM (512 nodes each) by showing Nietzsche texts.

• A sequence of 600901 characters consisting of 59 symbols (uppercase, lowercase, special characters).

Sample of training data

SUPPOSING that Truth is a woman--what then? Is there not ground for suspecting that all philosophers, in so far as they have been dogmatists, have failed to understand women--that the terrible seriousness and clumsy importunity with which they have usually paid their addresses to Truth, have been unskilled and unseemly methods for winning a woman? Certainly she has never allowed herself to be won
Text Modeling Experiment

- The training runs for a few hours on a Nvidia high end GPU (Tesla K40m).
- At start, the net knows only a few words, but picks up the vocabulary rather soon.

![Graph showing learning curve over epochs]

Epoch 1

Epoch 25

Epoch 3

It is the more the the the and the the and of the hos an the the and and the the the the the an the the the and the the ant an the and on the the he the the he hor an the the hore the the the he the he ans ante an the ankle the and and of the the hor and the the the the he the the the the he the the he the the he the the he the the he the the the and of an the the he the the the the the the he

Artists if they happened and for the concurred and man actored of shere all their accorition of the world the balirition and in the all of the prose goominity and in no berigation of the expores in a dondicted and for the forter prosoment that a conducted and of the menters of the soul of the dost and the for the

Manifold was not the little have a strong and contrary, his can be true to be a great need in the will to prove and consequence in short, something hably on the development of the intellectual and truth, and consequently, a little truth and possible all the higher things than the mastering sense of the servant of the envigh of the sense of the serve (and who has the goal it of this fantastic and the di
Text Modeling Experiment

• Let’s do the same thing for Finnish text: All discussions from Suomi24 forum are released for public.

• The message is nonsense, but syntax close to correct: A foreigner can not tell the difference.
Fake Chinese Characters

http://tinyurl.com/no36azh
EXAMPLES
Age / Gender / Expression Recognition

- TUT age estimation demo is an example of modern computer vision
- System estimates the age in real time
- Trained using a 500 K image database
- Average error ±3 years
Deep Net Learns to Play

- Mnih et al. (Google Deepmind, 2015) trained a network to play computer games
  - Better than human in many classic 1980’s games: 
    - Pinball
    - Pong
    - Space Invaders.
Computer and Logical Reasoning

• Logical reasoning is considered as a humans-only skill

• In this example, the computer was shown 1,000 question and answers

• In all 10 categories, the computer answers with > 95% accuracy (except Task 7: 85%)

From Image to Text

"man in black shirt is playing guitar."
"construction worker in orange safety vest is working on road."
"two young girls are playing with lego toy."
"boy is doing backflip on wakeboard."

From Video to Text

https://www.youtube.com/watch?v=8BFzu9m52sc

a group of people walking down a street with a car
Artistic Style Transfer

Check out Prisma App
Generative Adversarial Networks

- Recent work on generative networks has produced impressive results.
- Two networks are competing: one generating fake samples, the other trying to detect fakes.
GAN for Faces

- State of the art generates extremely realistic face images.
- Still, each is far from any of the training samples.
- [https://youtu.be/XOxxPcy5Gr4](https://youtu.be/XOxxPcy5Gr4)
To Conclude...

- During the last ten years, the landscape of artificial intelligence has reached a new level of maturity:
  - **Infrastructure** has been built to allow low cost access to high-performance computing.
  - **Publicity** of the results has become a standard model in dissemination of the research results.
  - **Resources** have increased: Companies are extremely active in AI research, and aggressively headhunting for the best talents in the field.
  - **Methods** have been improved and computers are increasingly able to solve human-like tasks.