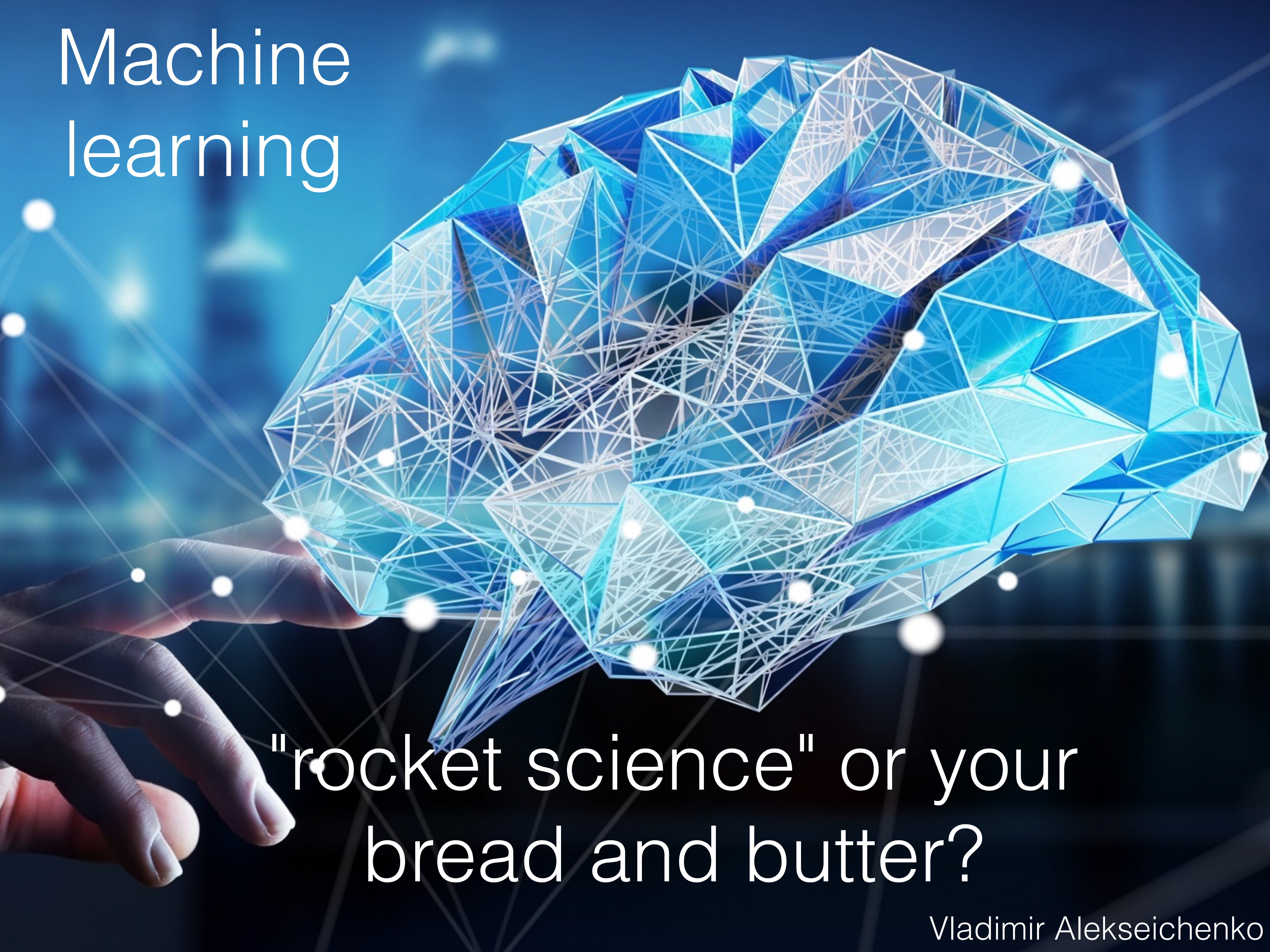
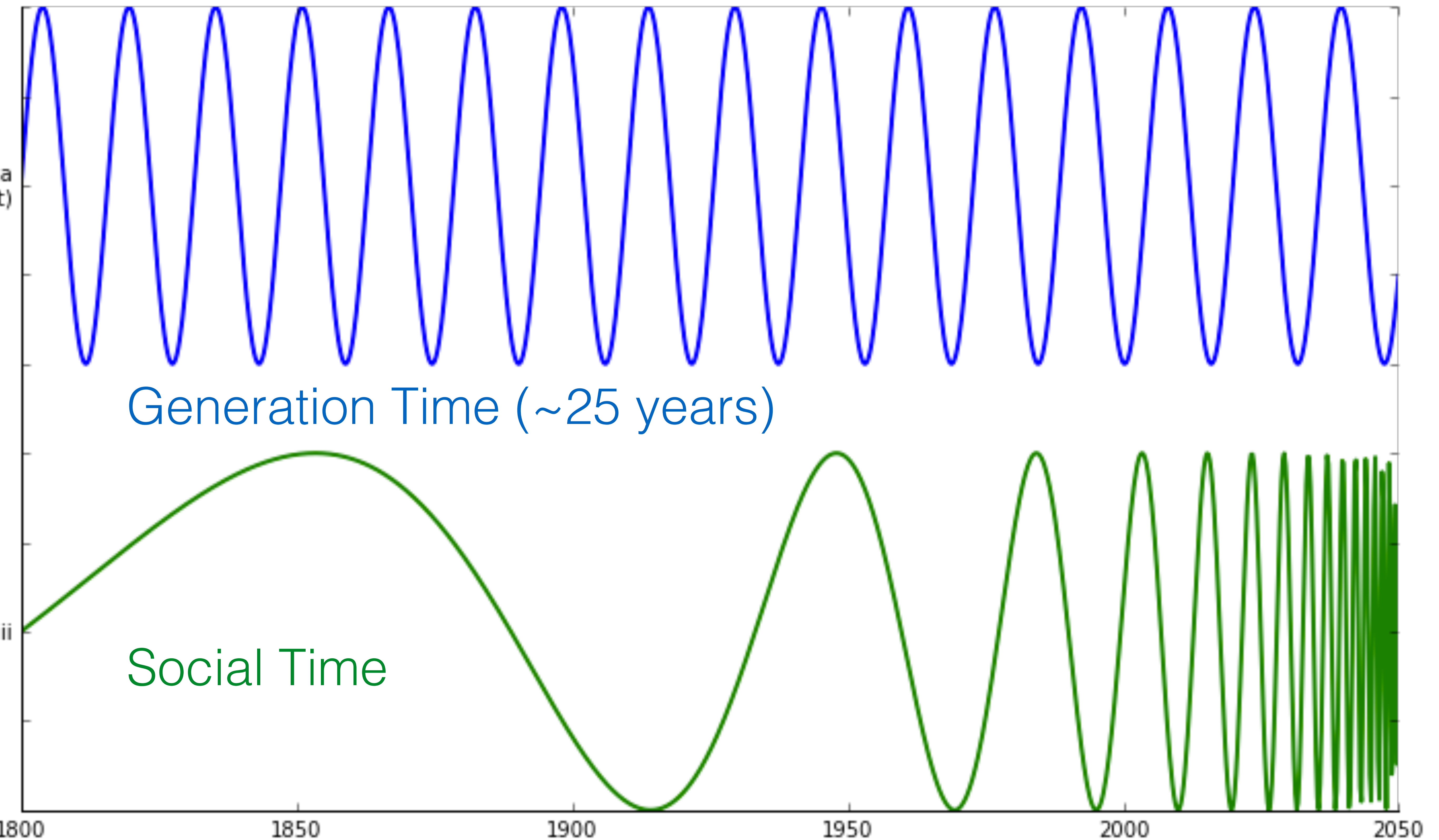
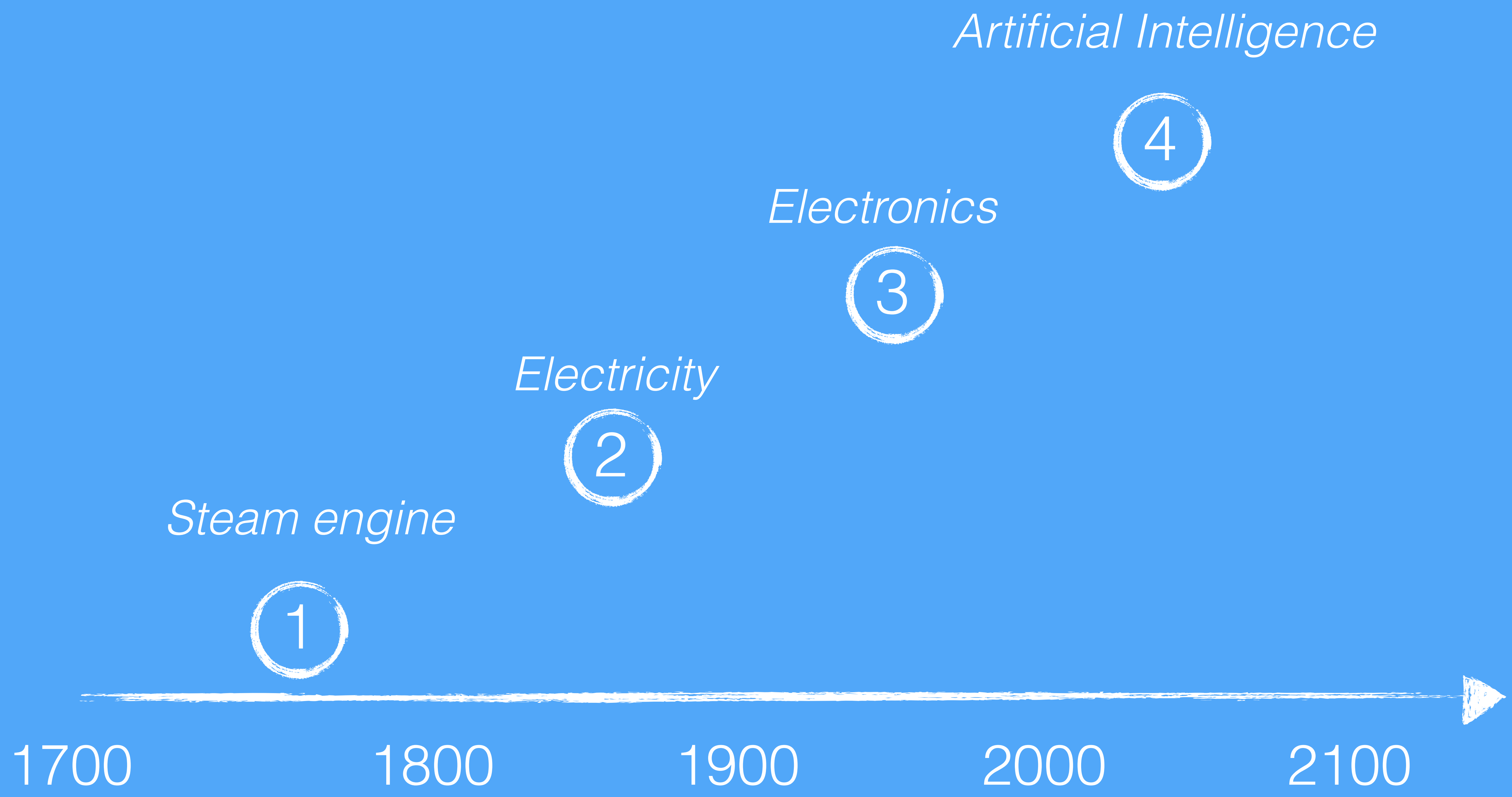

Machine learning
"rocket science" or your bread and butter?
Vladimir Alekseichenko

Changes in Time



The Fourth Industrial Revolution



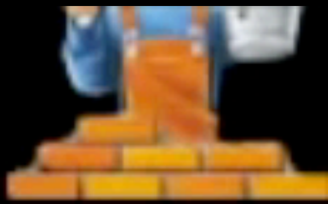
First drone delivery

https://www.youtube.com/watch?v=l96-OLmEpB8

Apple-Picking Robot

https://www.technologyreview.com/s/604303/apple-picking-robot-prepares-to-compete-for-farm-jobs/

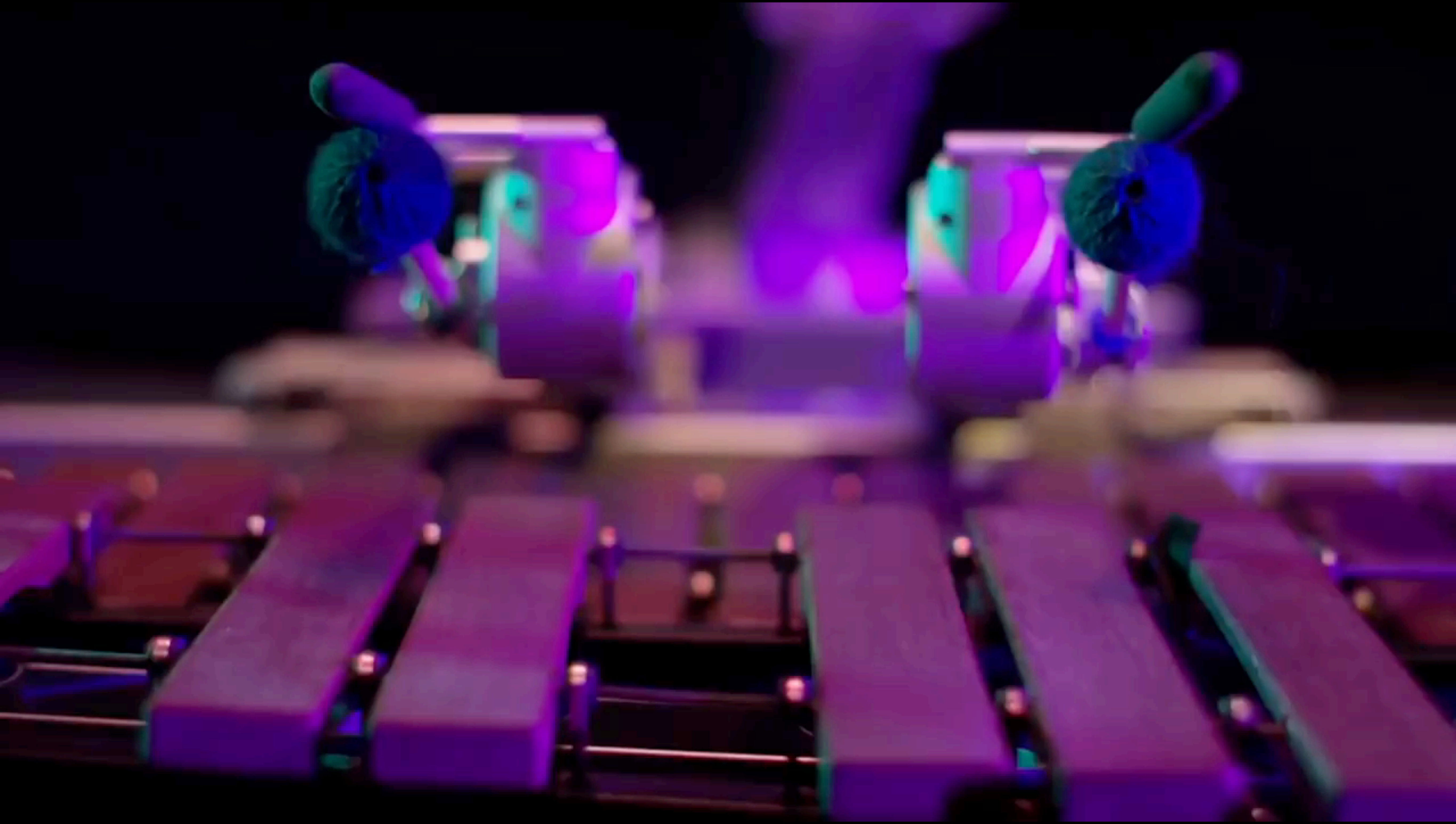
The robot can build an entire house in just **two days**



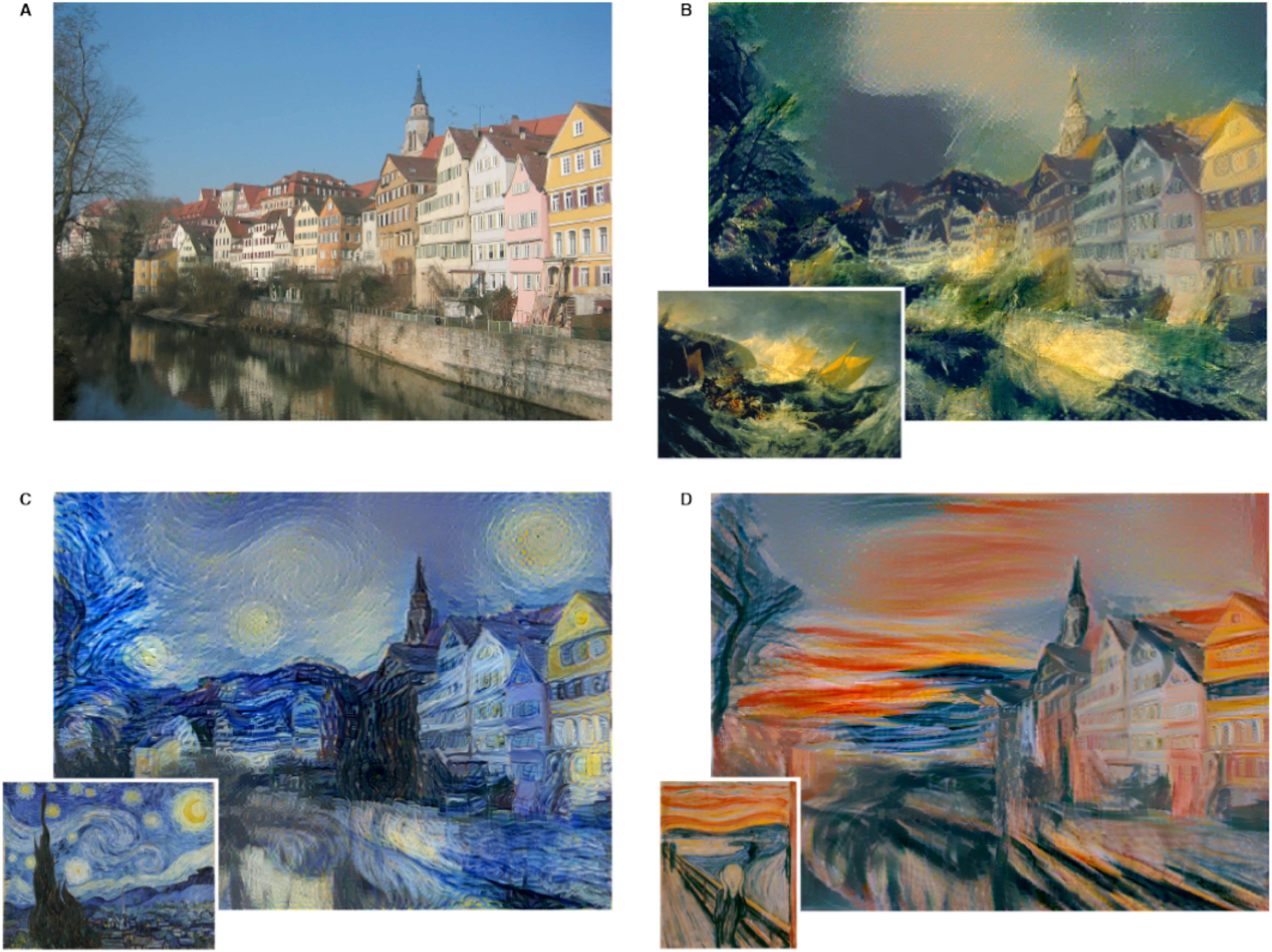
https://www.youtube.com/watch?v=S6UPGk8EFWE

Johann Sebastian **Bach**

Shimon Robot



https://www.shimonrobot.com/



A
B
C
D

+

=

https://research.googleblog.com/2016/02/exploring-intersection-of-art-and.html

Will a robot take your job?

11 September 2015 | Technology

Share

Type your job title into the search box below to find out the likelihood that it could be automated within the next two decades.

About 35% of current jobs in the UK are at high risk of computerisation over the following 20 years, according to a study by researchers at Oxford University and Deloitte.

I am a...

Can't find your job? **Browse the full list**

Find out my automation risk >

http://www.bbc.com/news/technology-34066941

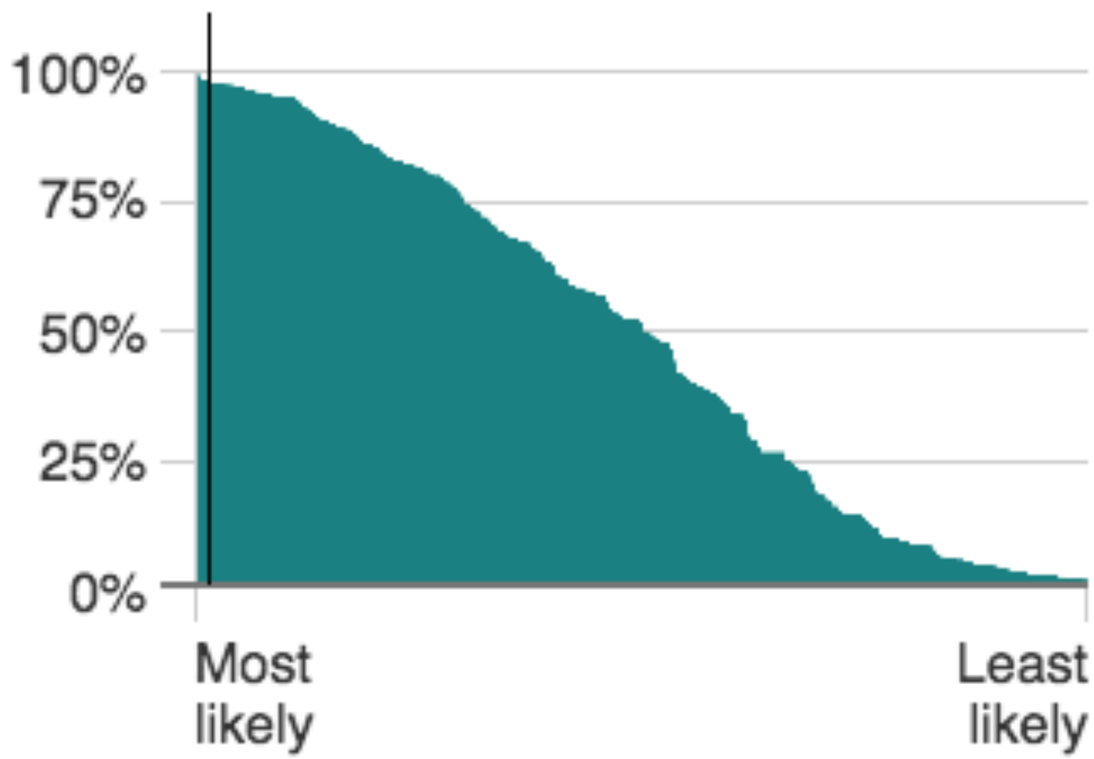
Financial accounts managers

Likelihood of automation?

It's quite likely (98%)

How this compares with other jobs:

4th of 366



Share my result

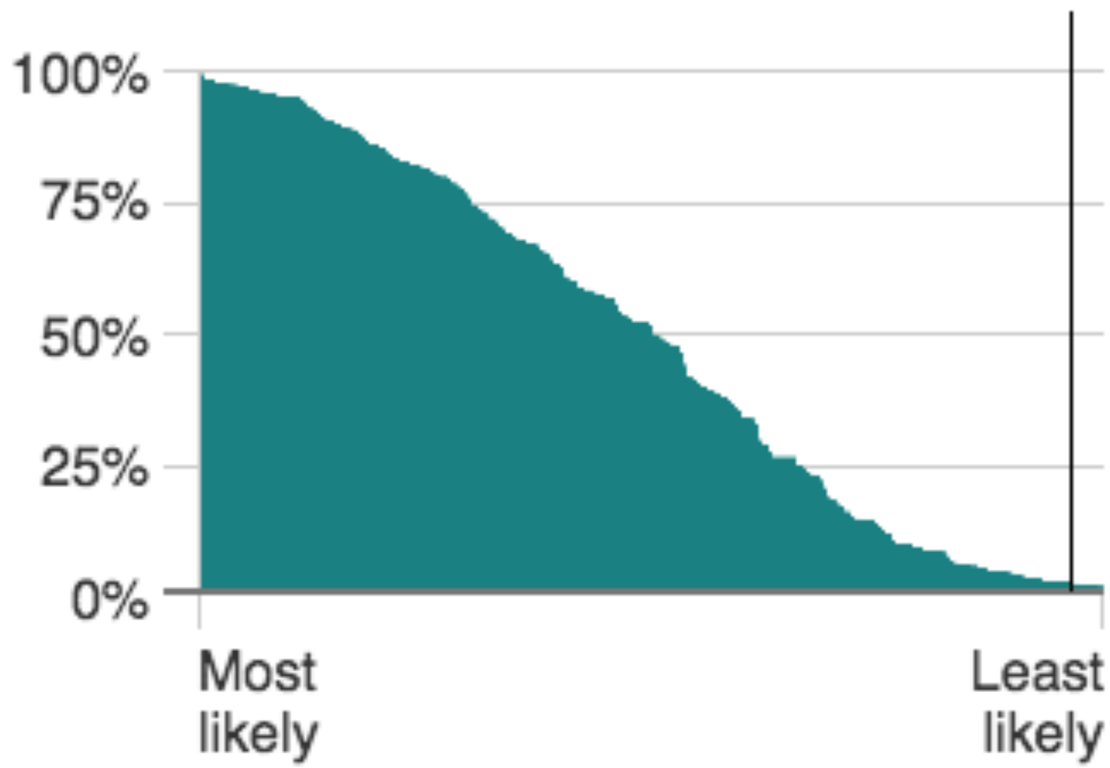
IT business analysts, architects and systems designers

Likelihood of automation?

It's quite unlikely (1%)

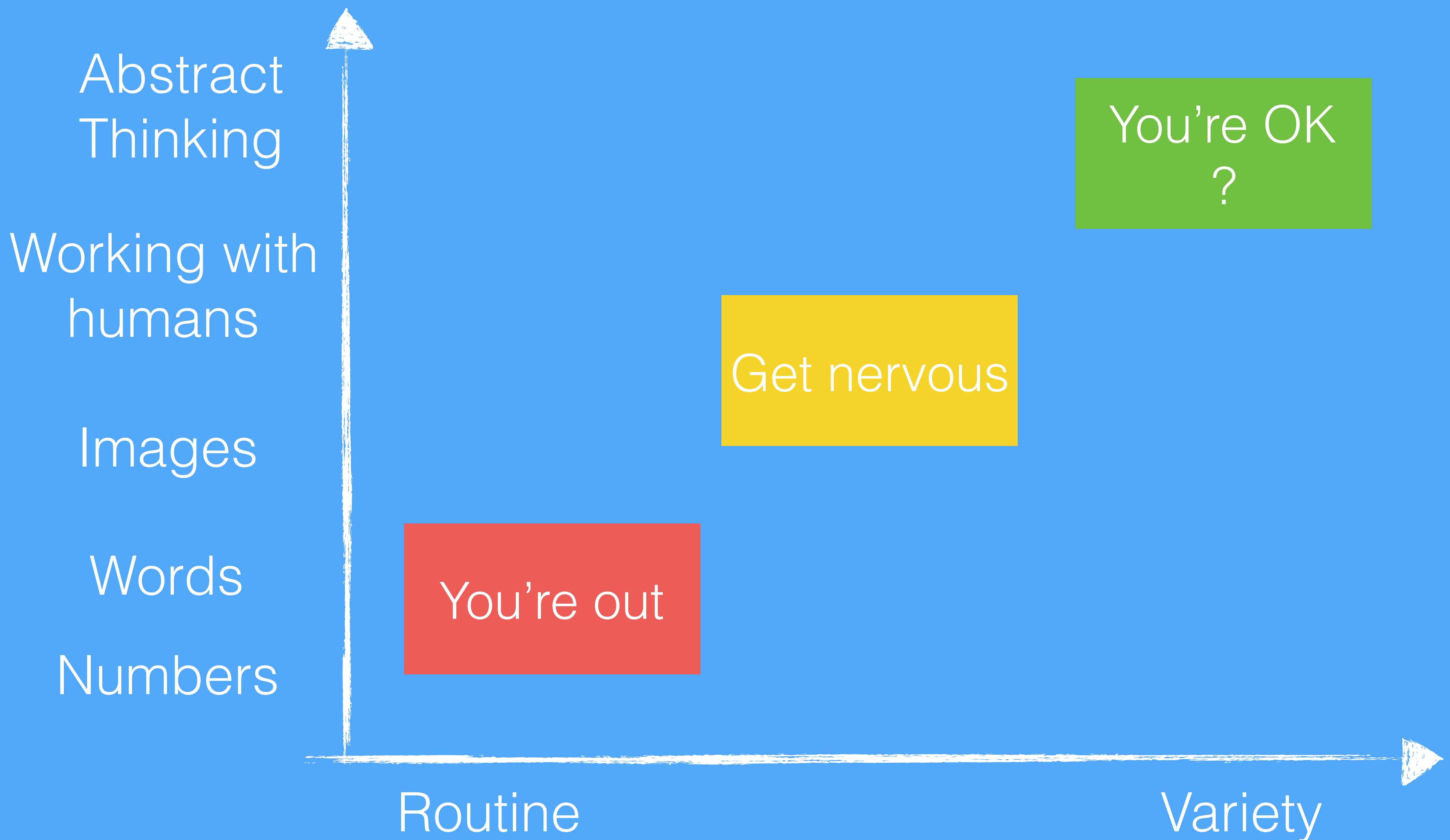
How this compares with other jobs:

353rd of 366

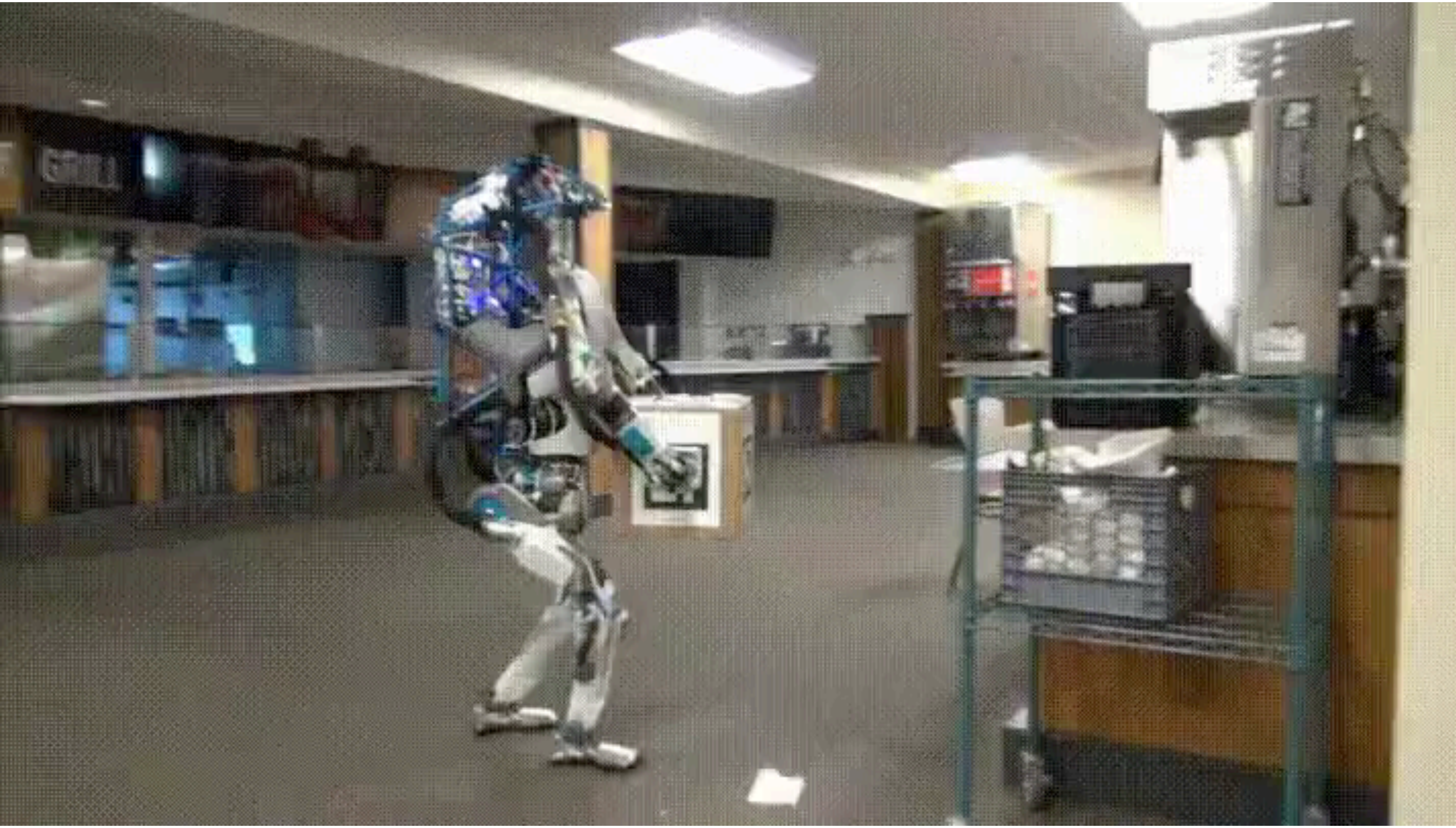


Share my result

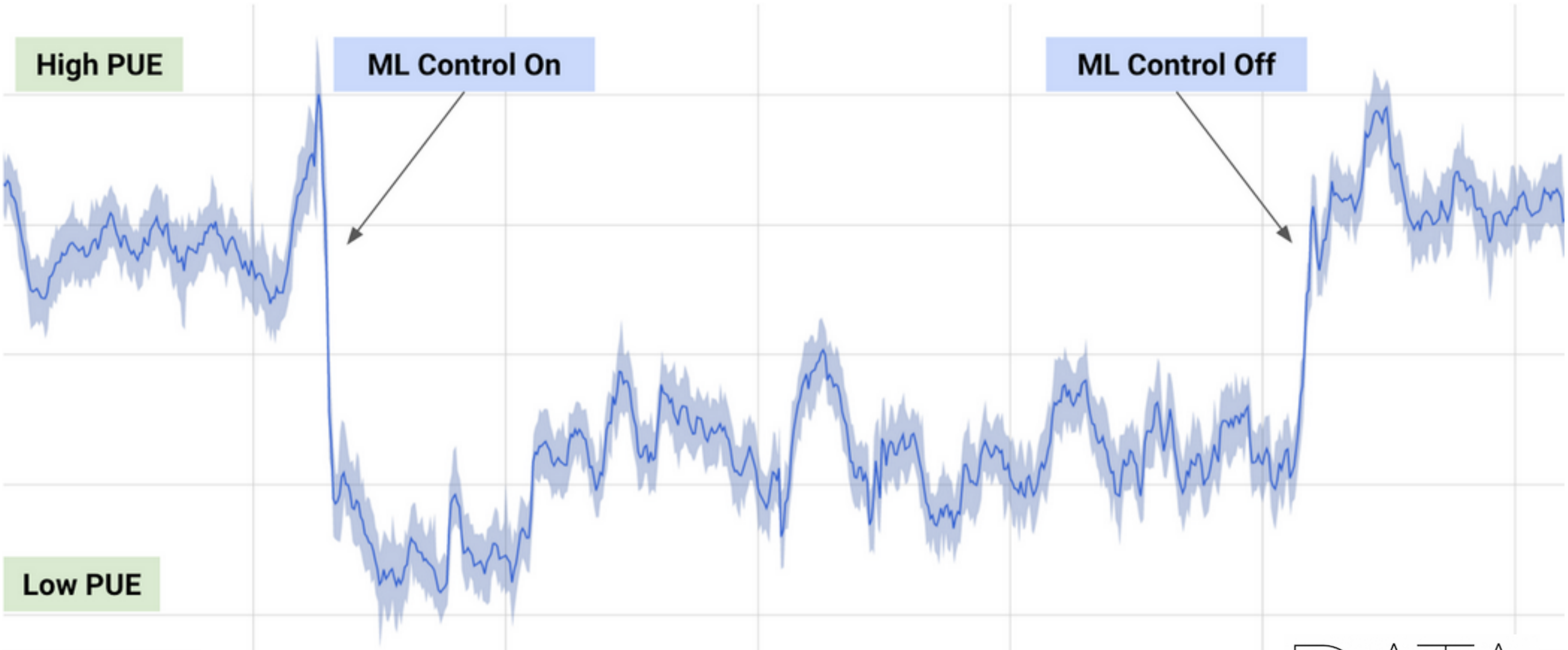
How safe is my job?



Hatsune Miku



Reduces Google Data Centre Cooling Bill by 40%



https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/

DATA WORKSHOP

Fraud Detection

PayPal

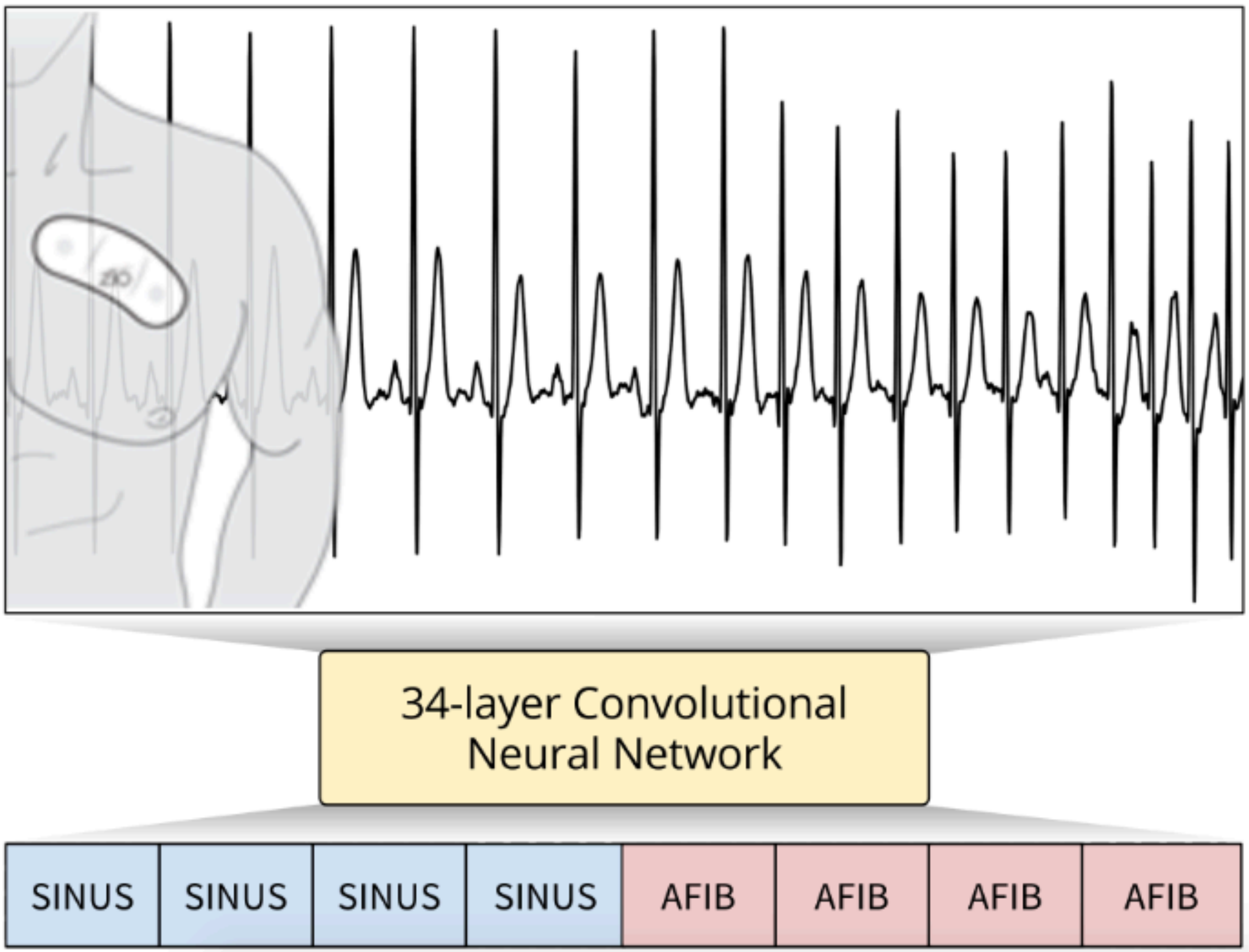
$2.35 billions
savings

only 0.32% of revenue

industry averages 1.32%
Federal Reserve Payments Study

https://www.technologyreview.com/s/545631/how-paypal-boosts-security-with-artificial-intelligence/

Stanford
University



exceed the average cardiologist performance

https://arxiv.org/pdf/1707.01836.pdf

DATA
WORKSHOP

The first mobile phone call in New York City in **1973**

This is a Motorola cellular portable telephone.

First in the market and first in use. It's the best-selling hand-held cellular portable on earth.

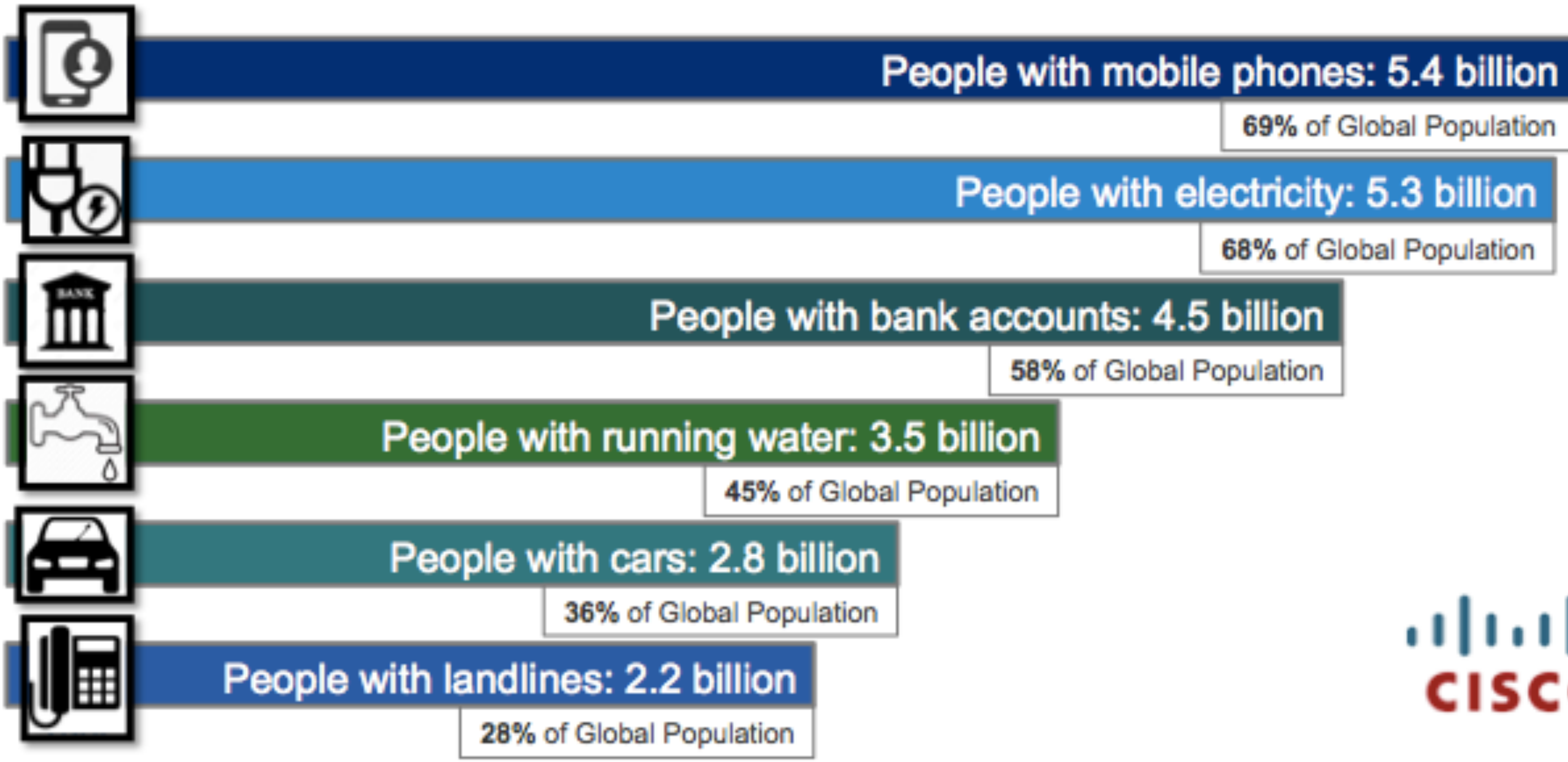
Take it to work, to play, to lunch and still keep up with your customers, your suppliers, your life.

MOTOROLA

Advanced Electronics for a More Productive World.

Mobile Growth Continues Through 2020

By 2020, more people will have mobile phones than electricity at home

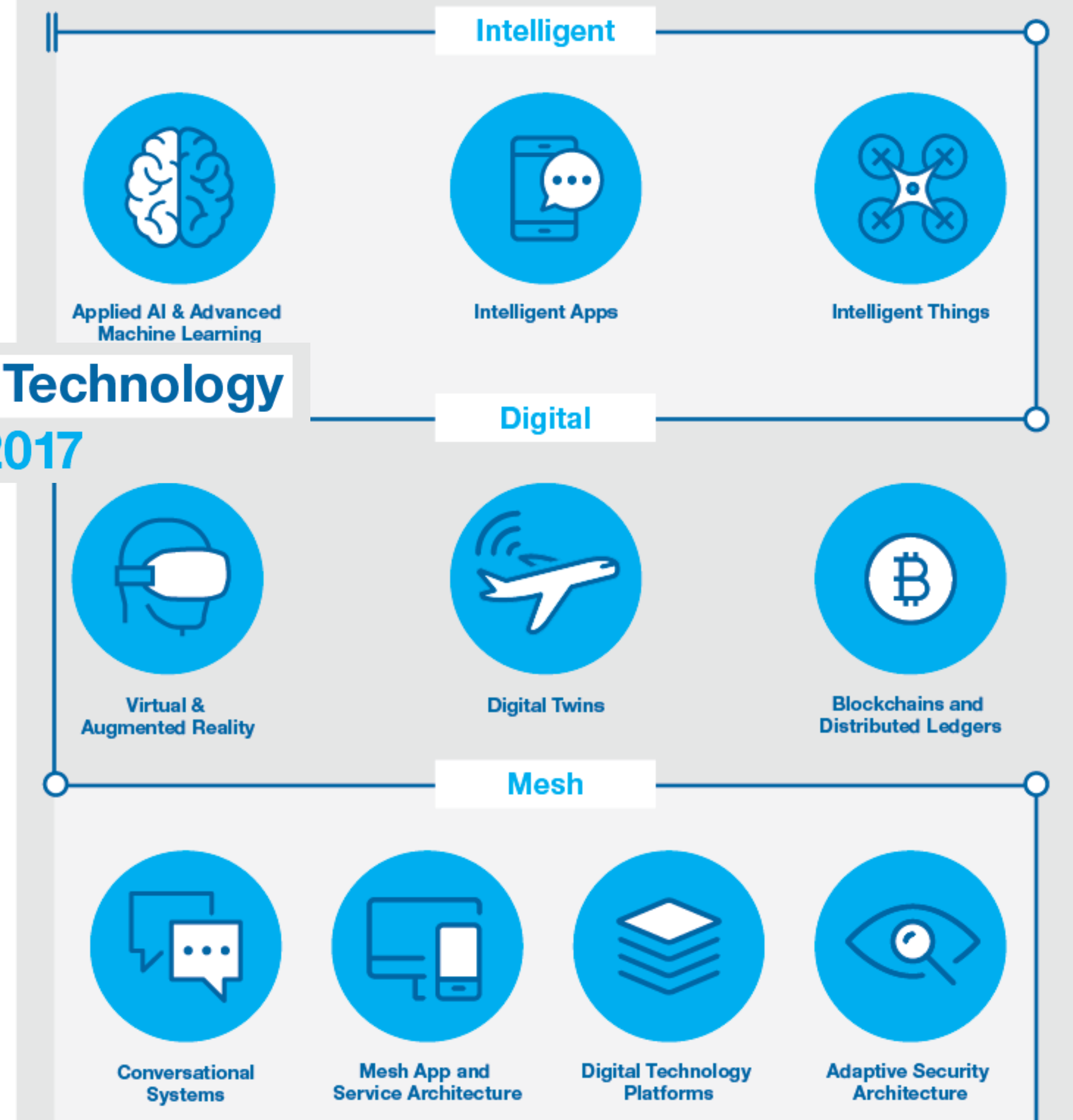


http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

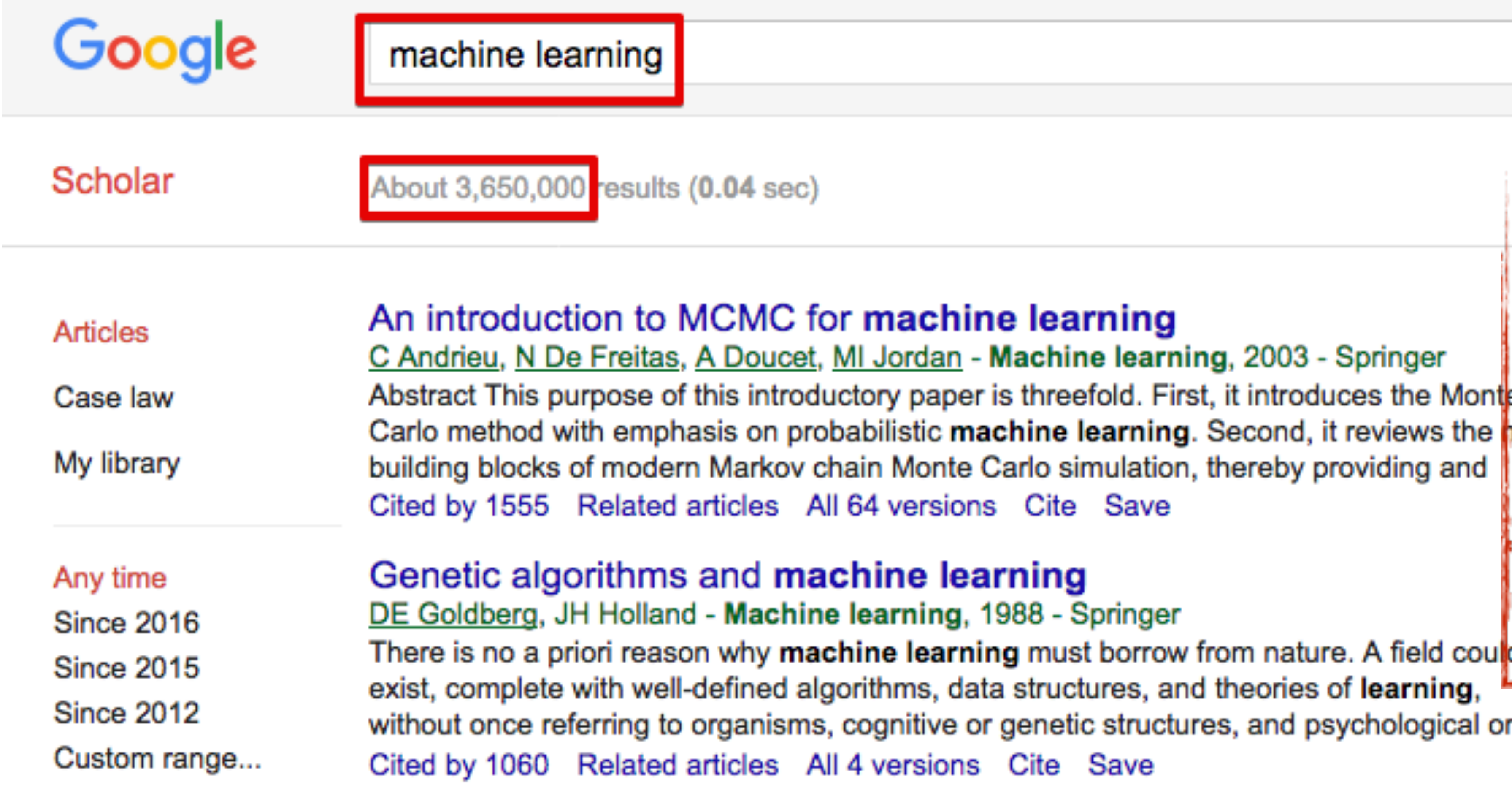
In next 10 years

- Are you excited about ML?
- What? Aaa, boring…

Top 10 Strategic Technology Trends 2017



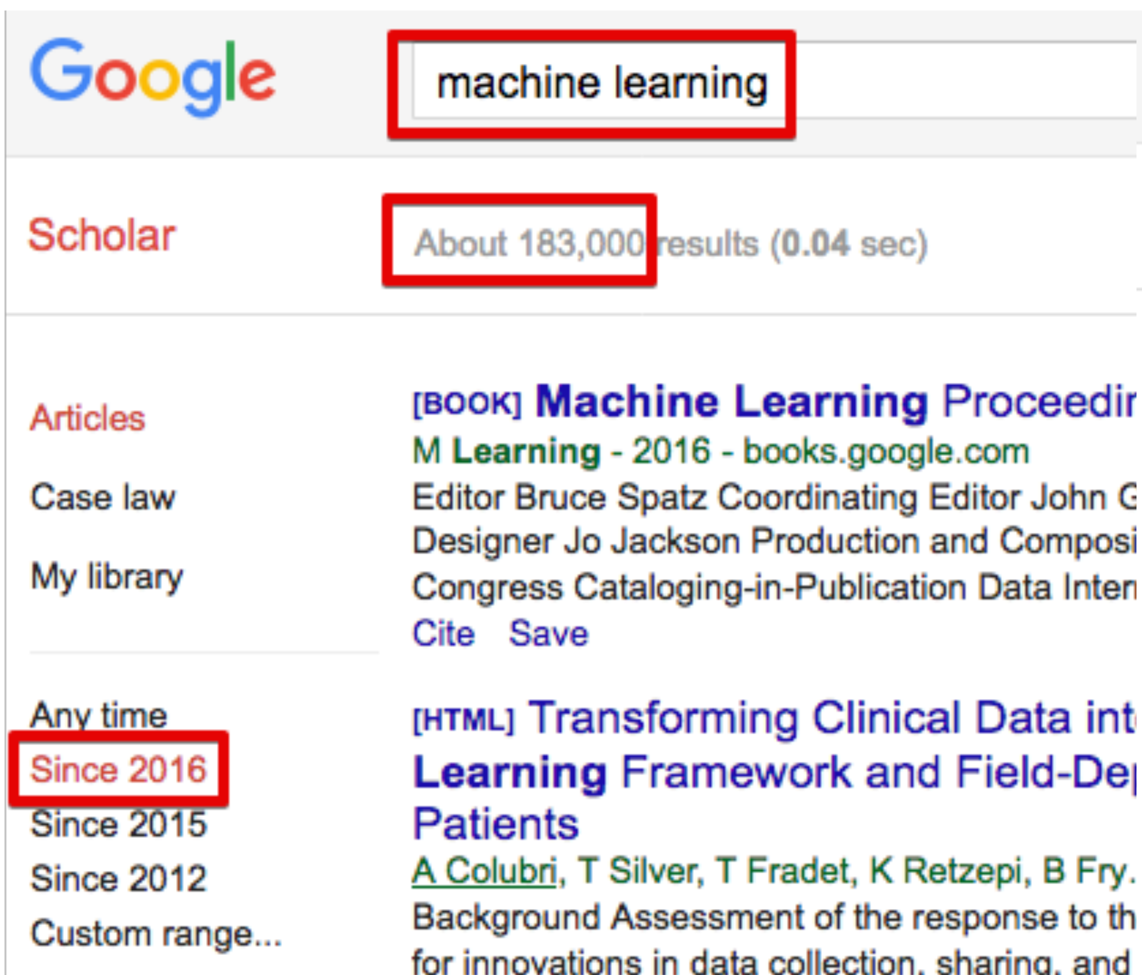
Gartner

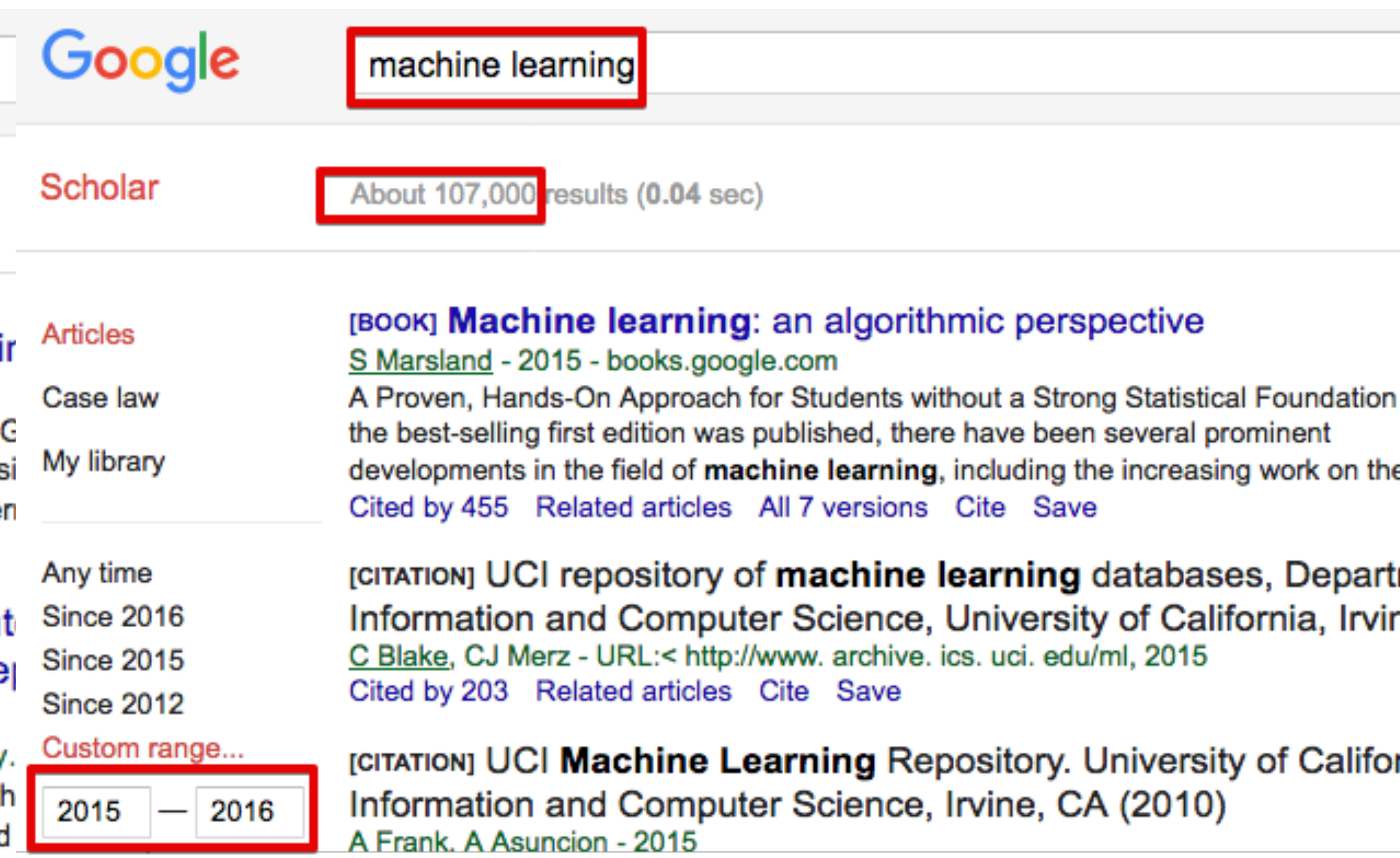


10min per one

36 500 000 mins

~70 years





$$\frac{\partial L}{\partial Y} : dY = \frac{\partial L}{\partial Y} : \mathcal{L}(dX) \triangleq \mathcal{L}^*\left(\frac{\partial L}{\partial Y}\right) : dX \Rightarrow \mathcal{L}^*\left(\frac{\partial L}{\partial Y}\right) = \frac{\partial L \circ f}{\partial X}$$

$$L \circ f(X + dX) - L \circ f(X) = \frac{\partial L \circ f}{\partial X} : dX + \mathrm{O}(\|dX\|^2)$$

$$L(Y + dY) - L(Y) = \frac{\partial L}{\partial Y} : dY + \mathrm{O}(\|dY\|^2)$$

$$\begin{aligned}
&=(DV^\top : dX - U^\top D : d\Sigma - VD^\top U\Sigma : dV) + \frac{\partial L}{\partial \Sigma} : (U^\top dX \\
&+ \frac{\partial L}{\partial V} : \{2V\left(K^\top \circ (\Sigma^\top U^\top dXV)_{sym}\right)\} \\
&=DV^\top : dX + \left(\frac{\partial L}{\partial \Sigma} - U^\top D\right) : (U^\top dXV)_{diag} + \\
&+ \left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right) : \{2V\left(K^\top \circ (\Sigma^\top U^\top dXV)_{sym}\right)\} \\
&=DV^\top : dX + \left(\frac{\partial L}{\partial \Sigma} - U^\top D\right)_{diag} : (U^\top dXV) + \\
&+ 2V^\top\left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right) : \left(K^\top \circ (\Sigma^\top U^\top dXV)_{sym}\right) \\
&=DV^\top : dX + U\left(\frac{\partial L}{\partial \Sigma} - U^\top D\right)_{diag} V^\top : dX + \\
&+ 2\left(K^\top \circ \left(V^\top\left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right)\right)\right)_{sym} : \Sigma^\top U^\top dXV \\
&=DV^\top : dX + U\left(\frac{\partial L}{\partial \Sigma} - U^\top D\right)_{diag} V^\top : dX + \\
&+ 2U\Sigma\left(K^\top \circ \left(V^\top\left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right)\right)\right)_{s}
\end{aligned}$$

$$\frac{\partial L \circ f}{\partial X} = U\left\{\left(\tilde{K}^\top \circ \left(U^\top \frac{\partial L}{\partial U}\right)\right) + \left(\frac{\partial L}{\partial \Sigma}\right)_{diag}\right\} U^\top$$

$$\frac{\partial L \circ f}{\partial \Sigma} = 2\Sigma g'(\Sigma^\top \Sigma + \epsilon I)V^\top \left(\frac{\partial L}{\partial C}\right)_{sym} V$$

$$\begin{aligned}
\frac{\partial L}{\partial C} : dC &= \frac{\partial L}{\partial C} : \left\{2\left(dVg(\Sigma^\top \Sigma + \epsilon I)V^\top\right)_{sym} + 2\left(Vg'(\Sigma^\top \Sigma + \epsilon I)\Sigma^\top d\Sigma V^\top\right)_{sym}\right\} \\
&=2\left(\frac{\partial L}{\partial C}\right)_{sym} : (dVg(\Sigma^\top \Sigma + \epsilon I)V^\top) + 2\left(\frac{\partial L}{\partial C}\right)_{sym} : (Vg'(\Sigma^\top \Sigma + \epsilon I)\Sigma^\top d\Sigma V^\top) \\
&=2\left\{\left(\frac{\partial L}{\partial C}\right)_{sym} Vg(\Sigma^\top \Sigma + \epsilon I)\right\} : dV + 2\left\{\Sigma g'(\Sigma^\top \Sigma + \epsilon I)V^\top \left(\frac{\partial L}{\partial C}\right)_{sym} V\right\} : d\Sigma
\end{aligned}$$

$$\begin{aligned}
p(I_i|B_i) &= p(I_{\Lambda_{B_i}}, I_{\overline{\Lambda_{B_i}}}|B_i) = p(I_{\Lambda_{B_i}}|B_i)p(I_{\overline{\Lambda_{B_i}}}|B_i) \\
&= p(I_{\Lambda_{B_i}}|B_i)q(I_{\overline{\Lambda_{B_i}}}) = q(I_\Lambda)\frac{p(I_{\Lambda_{B_i}}|B_i)}{q(I_{\Lambda_{B_i}})},
\end{aligned}$$

Proposition 1 (SVD Variations). *Let $X = U\Sigma V^\top$ with $X \in \mathbb{R}^{m,n}$ and $m \geq n$ Σ possessing diagonal structure. Then*

$$d\Sigma = (U^\top dXV)_{diag}$$

$$\begin{aligned}
Y_{i,k,\tilde{x},\tilde{y}} &= \sum_{c=1}^{C} D_{i,c,\tilde{x},\tilde{y}} * G_{k,c} \\
&= \sum_{c=1}^{C} A^T \left[U_{k,c} \odot V_{c,i,\tilde{x},\tilde{y}}\right] A \\
&= A^T \left[\sum_{c=1}^{C} U_{k,c} \odot V_{c,i,\tilde{x},\tilde{y}}\right] A
\end{aligned}$$

$$dV = 2V\left(K^\top \circ (\Sigma^\top U^\top dXV)_{sym}\right)$$

$$K_{ij} = \begin{cases} \frac{1}{\sigma_i^2 - \sigma_j^2}, & i \neq j \\ 0, & i = j \end{cases}$$

Let $\Sigma_n \in \mathbb{R}^{n\times n}$ be the top n rows of Σ and consider the block decomposi $\mathbb{R}^{m\times n}$ and $dU_2 \in \mathbb{R}^{m\times m-n}$ and similarly $\frac{\partial L}{\partial U} = \left(\left(\frac{\partial L}{\partial U}\right)_1 \middle| \left(\frac{\partial L}{\partial U}\right)_2\right)$, whe $\mathbb{R}^{m\times m-n}$. Then

$$dU = \left(C\Sigma_n^{-1} \middle| -U_1\Sigma_n^{-1}C^\top U_2\right)$$

$$C = dXV - Ud\Sigma - U\Sigma dV^\top V$$

Consequently the partial derivatives are

$$\frac{\partial L \circ f}{\partial X} = DV^\top + U\left(\frac{\partial L}{\partial \Sigma} - U^\top D\right)_{diag} V^\top + 2U\Sigma\left(K^\top \circ \left(V^\top \left(\frac{\partial}{\partial V}\right.\right.\right.$$

$$\frac{\partial L}{\partial Y} : dY = \frac{\partial L}{\partial Y} : \mathcal{L}(dX) \triangleq \mathcal{L}^* \left(\frac{\partial L}{\partial Y}\right) : dX \Rightarrow \mathcal{L}^* \left(\frac{\partial L}{\partial Y}\right) = \frac{\partial L \circ f}{\partial X}$$

$$L \circ f(X + dX) - L \circ f(X) = \frac{\partial L \circ f}{\partial X} : dX + \mathrm{O}(\|dX\|^2)$$

$$p(I_i|B_i) = p(I_{\Lambda_{B_i}}, I_{\overline{\Lambda_{B_i}}}|B_i) = p(I_{\Lambda_{B_i}}|B_i)p(I_{\overline{\Lambda_{B_i}}}|B_i)$$
$$= p(I_{\Lambda_{B_i}}|B_i)q(I_{\overline{\Lambda_{B_i}}}) = q(I_\Lambda)\frac{p(I_{\Lambda_{B_i}}|B_i)}{q(I_{\Lambda_{B_i}})},$$

https://media.giphy.com/media/xT9DPz0PhxgyrzLg7S/giphy.gif

$$+ 2U\Sigma\left(K^\top \circ \left(V^\top \left(\frac{\partial L}{\partial V} - VD^\top U\Sigma\right)\right)\right)$$

$$\frac{\partial L \circ f}{\partial X} = U\left\{\left(\tilde{K}^\top \circ \left(U^\top \frac{\partial L}{\partial U}\right)\right) + \left(\frac{\partial L}{\partial \Sigma}\right)_{diag}\right\}U^\top$$

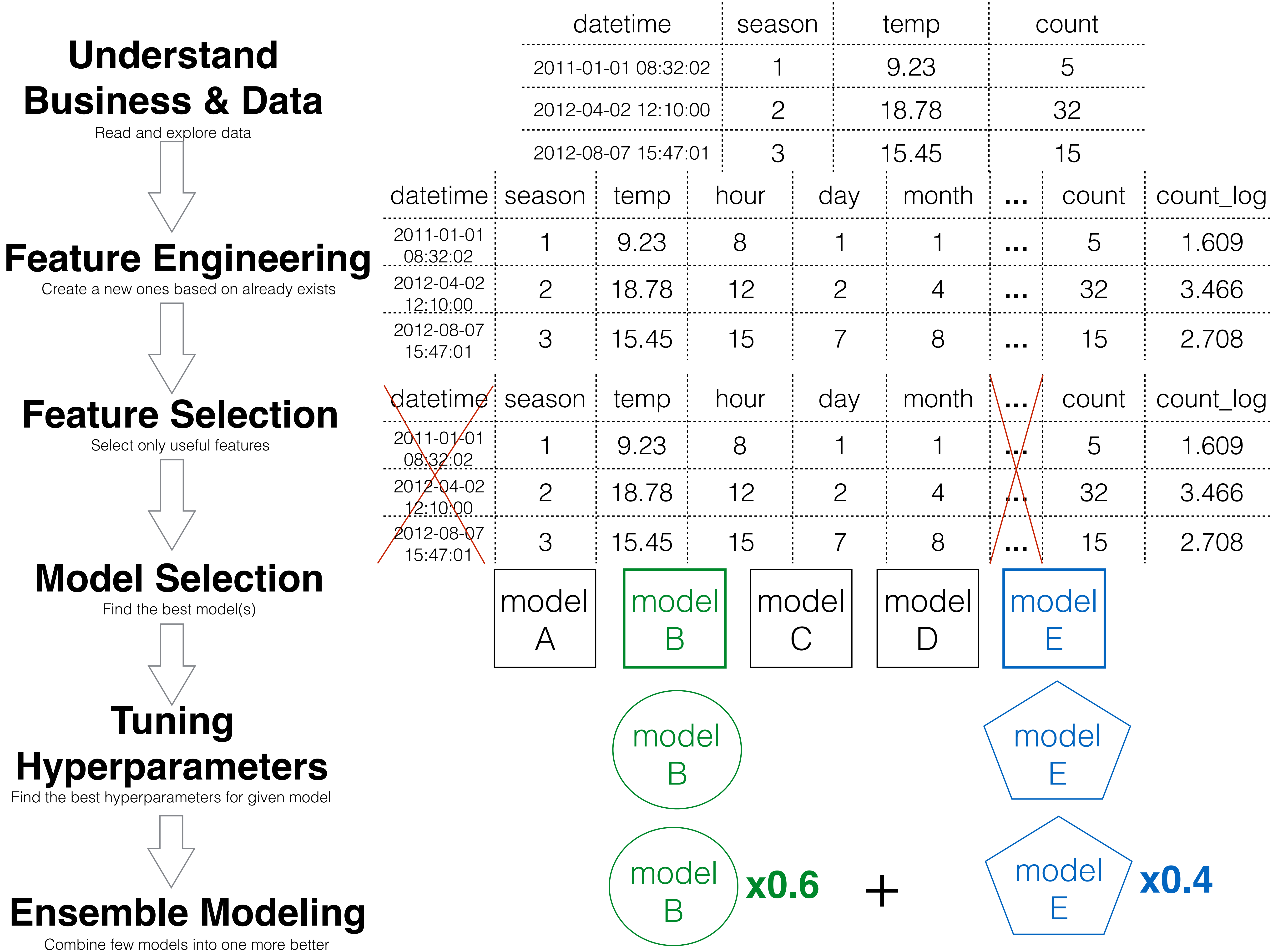
$$\frac{\partial L}{\partial C} : dC = \frac{\partial L}{\partial C} : \left\{2\left(dVg(\Sigma^\top\Sigma + \epsilon I)V^\top\right)_{sym} + 2\left(Vg'(\Sigma^\top\Sigma + \epsilon I)\Sigma^\top d\Sigma V^\top\right)_{sym}\right\}$$
$$= 2\left(\frac{\partial L}{\partial C}\right)_{sym} : \left(dVg(\Sigma^\top\Sigma + \epsilon I)V^\top\right) + 2\left(\frac{\partial L}{\partial C}\right)_{sym} : \left(Vg'(\Sigma^\top\Sigma + \epsilon I)\Sigma^\top d\Sigma V^\top\right)$$
$$= 2\left\{\left(\frac{\partial L}{\partial C}\right)_{sym} Vg(\Sigma^\top\Sigma + \epsilon I)\right\} : dV + 2\left\{\Sigma g'(\Sigma^\top\Sigma + \epsilon I)V^\top\left(\frac{\partial L}{\partial C}\right)_{sym} V\right\} : d\Sigma$$

Driver vs mechanic

Bike Sharing Demand

[Kaggle](#)

[Solution - github.com/dataworkshop](#)


Understand Business & Data
Read and explore data
datetime | season | temp | count
2011-01-01 08:32:02 | 1 | 9.23 | 5
2012-04-02 12:10:00 | 2 | 18.78 | 32
2012-08-07 15:47:01 | 3 | 15.45 | 15
Feature Engineering
Create a new ones based on already exists
datetime | season | temp | hour | day | month | ... | count | count_log
2011-01-01 08:32:02 | 1 | 9.23 | 8 | 1 | 1 | ... | 5 | 1.609
2012-04-02 12:10:00 | 2 | 18.78 | 12 | 2 | 4 | ... | 32 | 3.466
2012-08-07 15:47:01 | 3 | 15.45 | 15 | 7 | 8 | ... | 15 | 2.708
Feature Selection
Select only useful features
datetime | season | temp | hour | day | month | ... | count | count_log
2011-01-01 08:32:02 | 1 | 9.23 | 8 | 1 | 1 | ... | 5 | 1.609
2012-04-02 12:10:00 | 2 | 18.78 | 12 | 2 | 4 | ... | 32 | 3.466
2012-08-07 15:47:01 | 3 | 15.45 | 15 | 7 | 8 | ... | 15 | 2.708
Model Selection
Find the best model(s)
model A
model B
model C
model D
model E
Tuning Hyperparameters
Find the best hyperparameters for given model
model B
model E
Ensemble Modeling
Combine few models into one more better
model B
x0.6
+
model E
x0.4

Understand Business & Data

Read and explore data

datetime	season	temp	count
2011-01-01 08:32:02	1	9.23	5
2012-04-02 12:10:00	2	18.78	32
2012-08-07 15:47:01	3	15.45	15

Feature Engineering

Create a new ones based on already exists

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Feature Selection

Select only useful features

~~datetime~~	season	temp	hour	day	month	~~...~~	count	count_log
~~2011-01-01 08:32:02~~	1	9.23	8	1	1	~~...~~	5	1.609
~~2012-04-02 12:10:00~~	2	18.78	12	2	4	~~...~~	32	3.466
~~2012-08-07 15:47:01~~	3	15.45	15	7	8	~~...~~	15	2.708

Model Selection

Find the best model(s)

model A | model B | model C | model D | model E

Tuning Hyperparameters

Find the best hyperparameters for given model

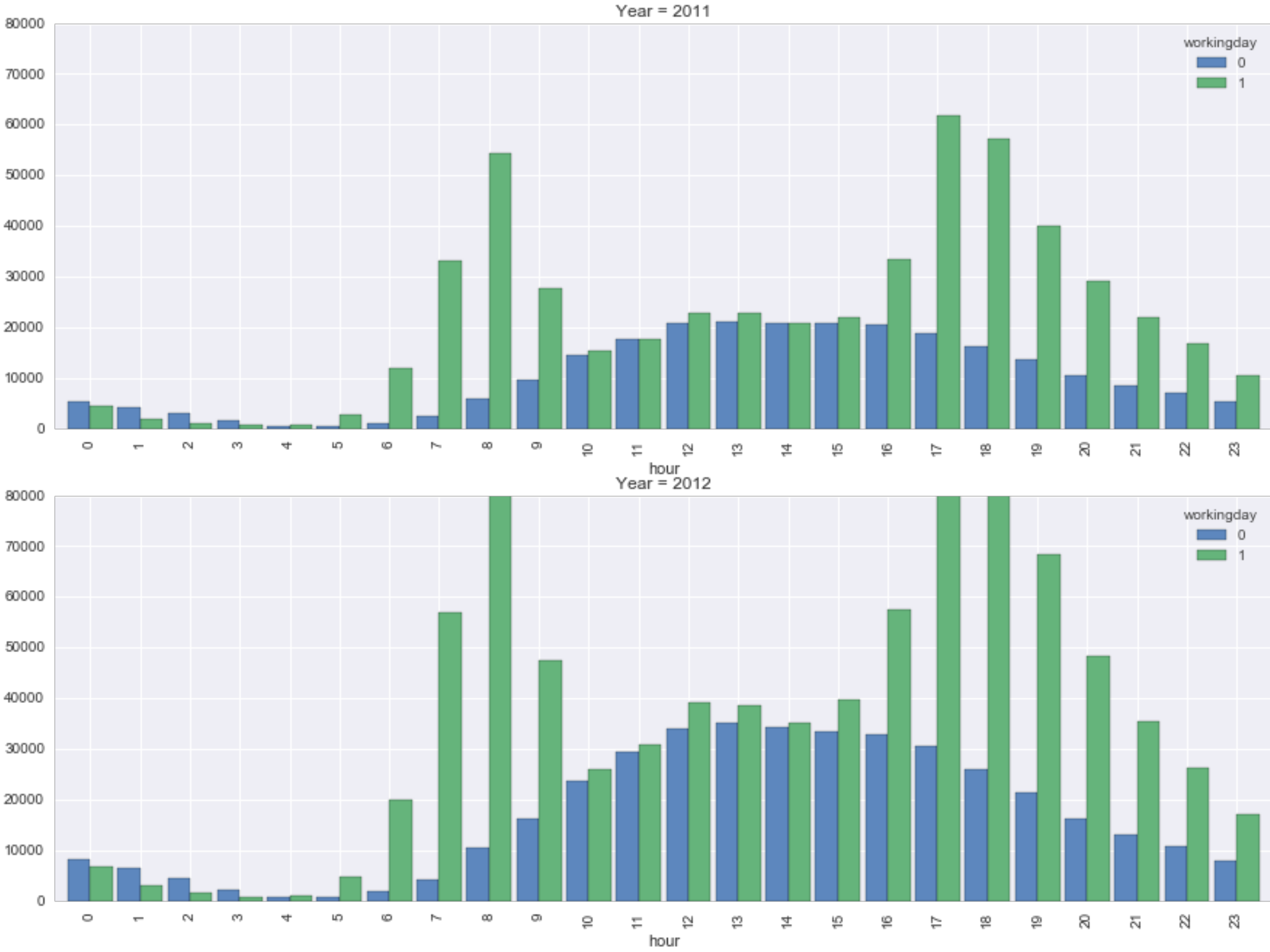
model B | model E

Ensemble Modeling

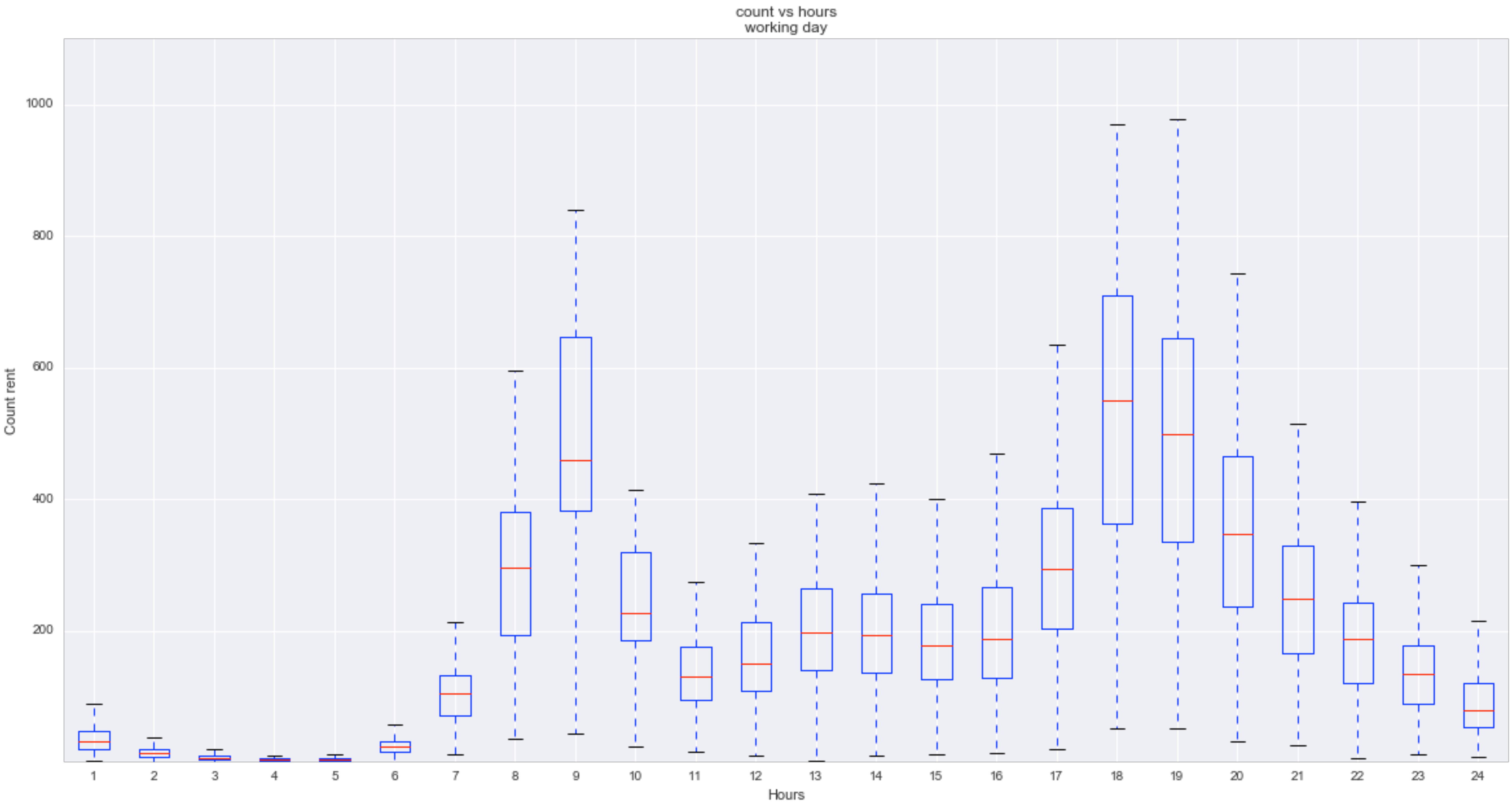
Combine few models into one more better

model B x0.6 + model E x0.4

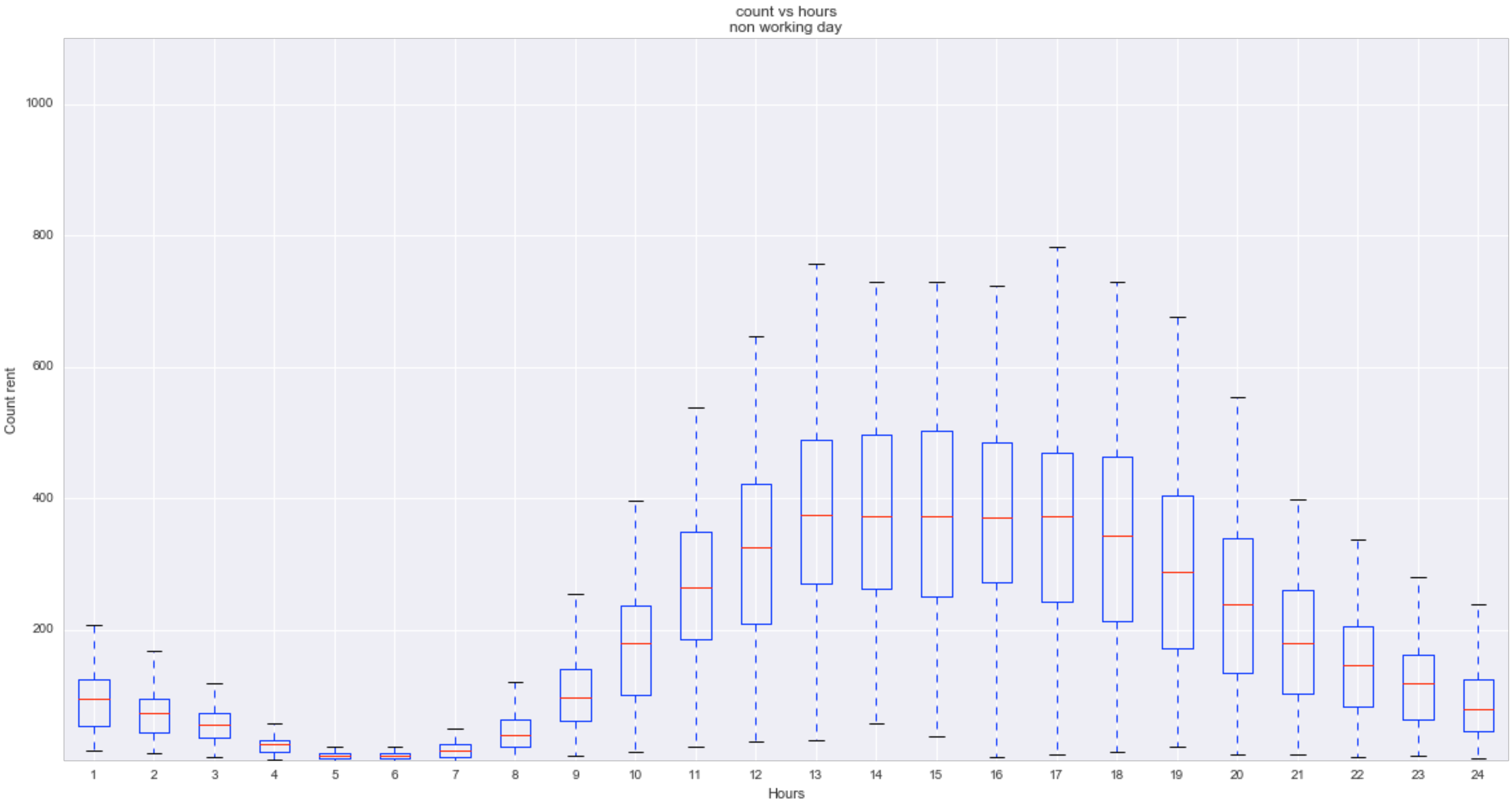
Understand Data and Business


Year = 2011
workingday
0
1
hour
Year = 2012
workingday
0
1
hour

Working days



Weekend



count vs hours
non working day
Count rent
1000
800
600
400
200
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Hours

Understand Business & Data

Read and explore data

datetime	season	temp	count
2011-01-01 08:32:02	1	9.23	5
2012-04-02 12:10:00	2	18.78	32
2012-08-07 15:47:01	3	15.45	15

Feature Engineering

Create a new ones based on already exists

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Feature Selection

Select only useful features

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Model Selection

Find the best model(s)

model A | model B | model C | model D | model E

Tuning Hyperparameters

Find the best hyperparameters for given model

model B | model E

Ensemble Modeling

Combine few models into one more better

model B x0.6 + model E x0.4

Feature Engineering

- **Continuous =>** from 1 to 100…
- **dates =>** day, month, year, hour, is weekend…
- **categorical** (red, green, white)
  - assign an unique ID (1, 2, 3)
  - create n-binary columns (is red? etc)
  - probability with target variable (if red 20% then…)

```python
def feature_engineering(data):
    data['year'] = data['datetime'].dt.year
    data['diff_year'] = data['year'] - 2010
    data['month'] = data['datetime'].dt.month
    data['day'] = data['datetime'].dt.day
    data['hour'] = data['datetime'].dt.hour
    data['minute'] = data['datetime'].dt.minute
    data['dayofweek'] = data['datetime'].dt.dayofweek
    data['weekofyear'] = data['datetime'].dt.weekofyear
    data['weekend'] = data.dayofweek.map(lambda x: int(x in [5,6]) )
    data['time_of_day'] = data['hour'].map(cat_hour)

    data['dayofyear'] = data['datetime'].dt.dayofyear
    data['day_'] = data[ ['year', 'dayofyear'] ].apply(lambda x: x['dayofyear'] + int(str(x['y

    data['rush_hour'] = data['datetime'].apply(lambda i: min([np.fabs(9-i.hour), np.fabs(20-i.
    data.loc[:,('rush_hour')] = data['datetime'].apply(lambda i: np.fabs(14-i.hour))
    data.loc[data['workingday'] != 0].loc[:,('rush_hour')] = 0

    data['holiday'] = data[['month', 'day', 'holiday', 'year']].apply(lambda x: (x['holiday'],
    data['holiday'] = data[['month', 'day', 'holiday']].apply(lambda x: (x['holiday'], 1)[x['m

    data['workingday'] = data[['month', 'day', 'workingday']].apply(lambda x: (x['workingday']
    data['peak'] = data[['hour', 'workingday']].apply(lambda x: (0, 1)[(x['workingday'] == 1 a
    data['sticky'] = data[['humidity', 'workingday']].apply(lambda x: (0, 1)[x['workingday'] =

    return data
```

Understand Business & Data

Read and explore data

datetime	season	temp	count
2011-01-01 08:32:02	1	9.23	5
2012-04-02 12:10:00	2	18.78	32
2012-08-07 15:47:01	3	15.45	15

Feature Engineering

Create a new ones based on already exists

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Feature Selection

Select only useful features

~~datetime~~	season	temp	hour	day	month	~~...~~	count	count_log
~~2011-01-01 08:32:02~~	1	9.23	8	1	1	~~...~~	5	1.609
~~2012-04-02 12:10:00~~	2	18.78	12	2	4	~~...~~	32	3.466
~~2012-08-07 15:47:01~~	3	15.45	15	7	8	~~...~~	15	2.708

Model Selection

Find the best model(s)

model A | model B | model C | model D | model E

Tuning Hyperparameters

Find the best hyperparameters for given model

model B | model E

Ensemble Modeling

Combine few models into one more better

model B x0.6 + model E x0.4

Feature selection

- Less means better (a simple model)
- Keep only the most valuable features (the ideal case - one :)
- Features (usually) depends on each other, be careful… (verify in an empiric way)
- Fewer features - faster solution

```
treshold = .95
sel = VarianceThreshold(threshold=(treshold * (1 - treshold)))
sel.fit_transform(train[feats], train['count'])
print(feats[~sel.get_support()])
```

```
['holiday' 'minute']
```

Variance

```
sel = SelectKBest(f_regression, 15)
sel.fit_transform(train[feats], train['count'])

print(feats[~sel.get_support()])
```

```
['minute' 'dayofweek']
```

Univariate

```
model = RandomForestRegressor()
sel = RFE(model, 15, step=1)
sel.fit_transform(train[feats], train['count'])

print(feats[~sel.get_support()])
```

```
['holiday' 'minute']
```

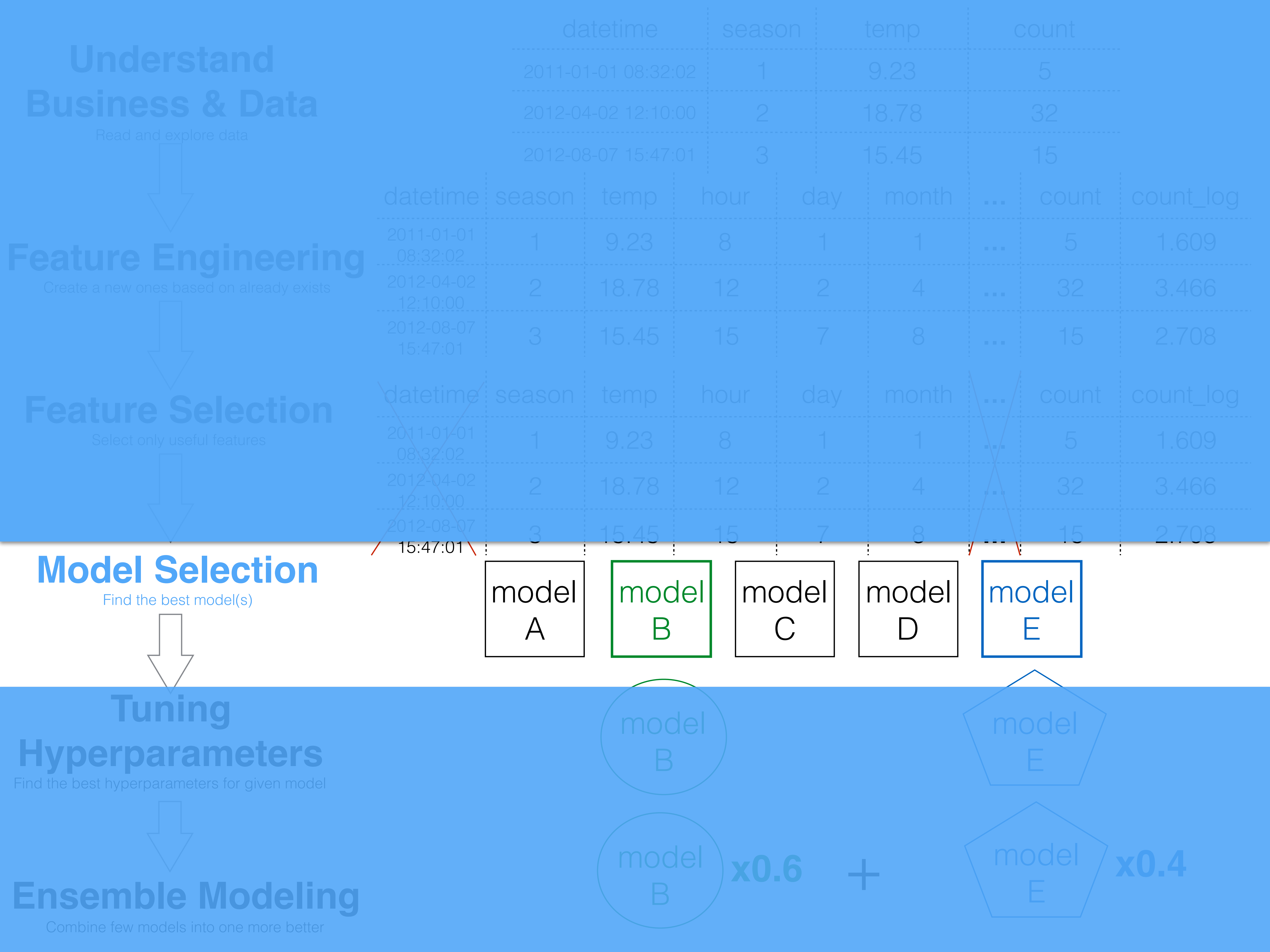
Recursive

xgbfir

```
xgb_rmodel = xgb.XGBRegressor().fit(train[feats], train['count'])
xgbfir.saveXgbFI(xgb_rmodel, feature_names=feats, OutputXlsxFile = 'features.xlsx')
```

A	B	C	D	E	F	G
Interaction	**Gain**	**FScore**	**wFScore**	**Average wFScore**	**Average Gain**	**Expected Gain**
hour	1054734195.06	322	126.5599853022	0.3930434326	3275572.03434783	760924362.285034
year	152732430	43	27.0461142752	0.6289794017	3551916.97674419	94928243.9262263
workingday	108062504.4	73	15.7825647621	0.2161995173	1480308.27945206	15599645.5009645
temp	94229934.3	36	21.1868454896	0.5885234858	2617498.175	67736321.1195572
atemp	75775010.47	29	16.3040602609	0.5622089745	2612931.39551724	62157912.1593818
month	42770622.1	37	20.5943413559	0.5566038204	1155962.75945946	17394990.8361565
dayofweek	25376066.7	46	14.4209994488	0.313499988	551653.623913043	8494132.2122359
humidity	23826037.09	34	15.0119419438	0.4415277042	700765.796764706	13932336.1630176
season	20955272.58	14	5.5503398861	0.396452849	1496805.18428571	12406814.115034
weather	18267125.7	24	19.1704023516	0.7987667647	761130.2375	12877953.5530039
weekofyear	17180242.5	19	9.8273929818	0.5172312096	904223.289473684	11365223.4477127
windspeed	927227.2	13	7.502572111	0.5771209316	71325.1692307692	551228.17539041
day	520283.8	6	0.9690428073	0.1615071345	86713.9666666667	71061.1130075326
holiday	275134.1	3	0.0731214404	0.0243738135	91711.3666666667	6470.5082674996

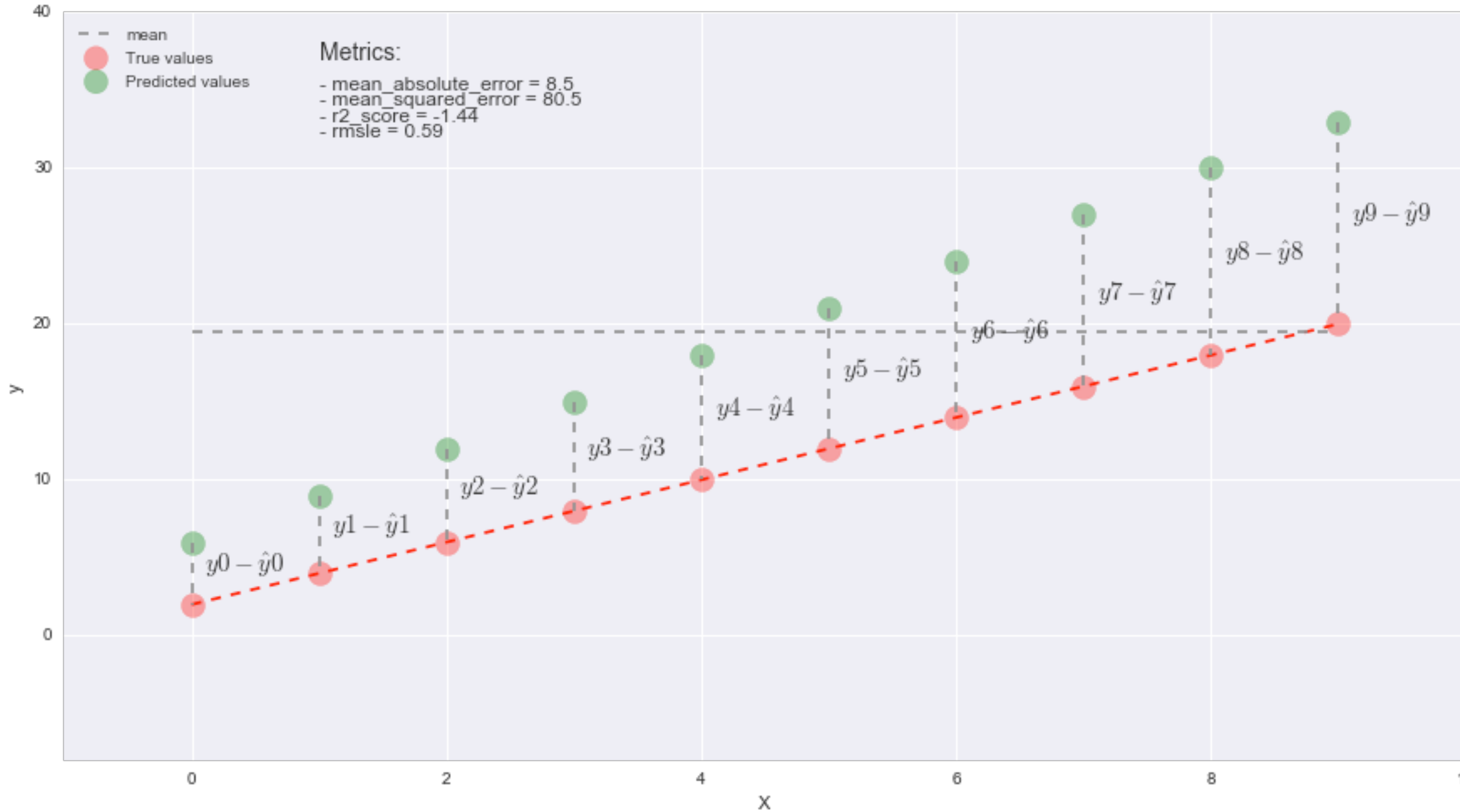
https://github.com/limexp/xgbfir


Understand Business & Data
Read and explore data
datetime season temp count
2011-01-01 08:32:02 1 9.23 5
2012-04-02 12:10:00 2 18.78 32
2012-08-07 15:47:01 3 15.45 15
Feature Engineering
Create a new ones based on already exists
datetime season temp hour day month ... count count_log
2011-01-01 08:32:02 1 9.23 8 1 1 ... 5 1.609
2012-04-02 12:10:00 2 18.78 12 2 4 ... 32 3.466
2012-08-07 15:47:01 3 15.45 15 7 8 ... 15 2.708
Feature Selection
Select only useful features
datetime season temp hour day month ... count count_log
2011-01-01 08:32:02 1 9.23 8 1 1 ... 5 1.609
2012-04-02 12:10:00 2 18.78 12 2 4 ... 32 3.466
2012-08-07 15:47:01 3 15.45 15 7 8 ... 15 2.708
Model Selection
Find the best model(s)
model A
model B
model C
model D
model E
Tuning Hyperparameters
Find the best hyperparameters for given model
model B
model E
Ensemble Modeling
Combine few models into one more better
model B x0.6 + model E x0.4

Model selection

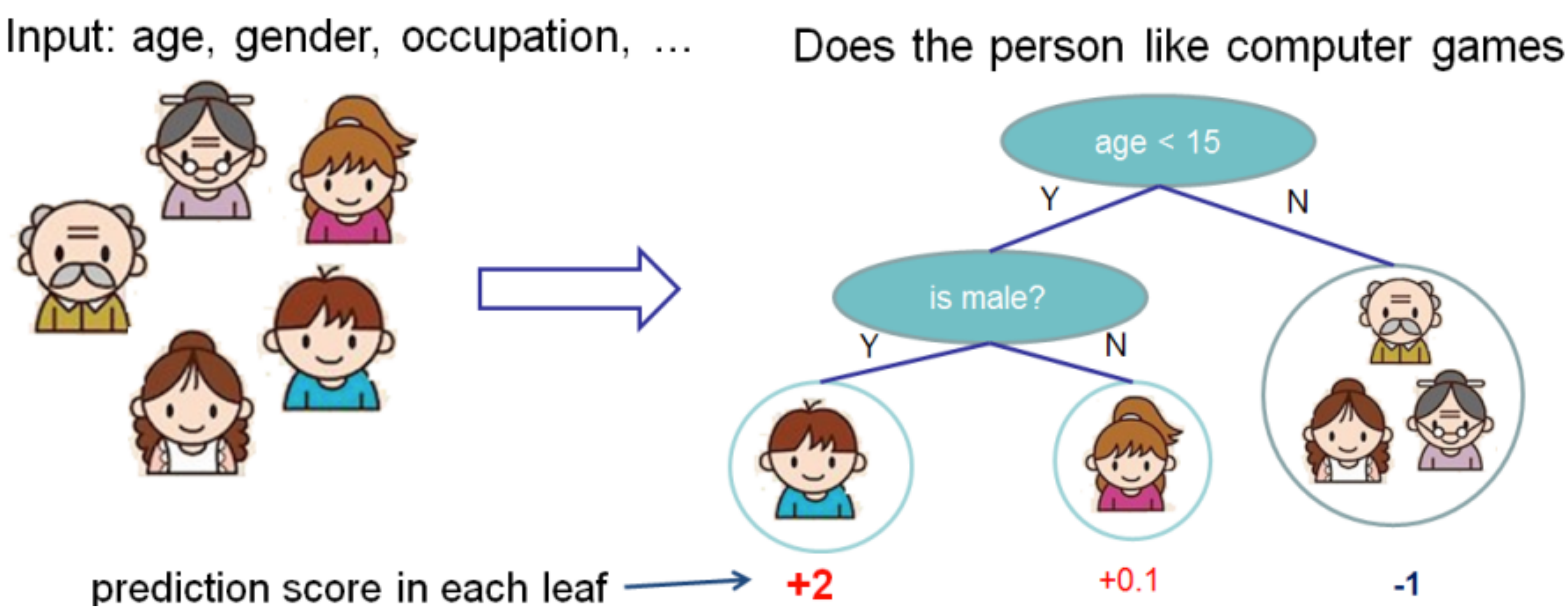
- Linear
- Decision Tree
- Random Forest
- Gradient Boosting
- Neural Network (CNN, RNN…)

Linear



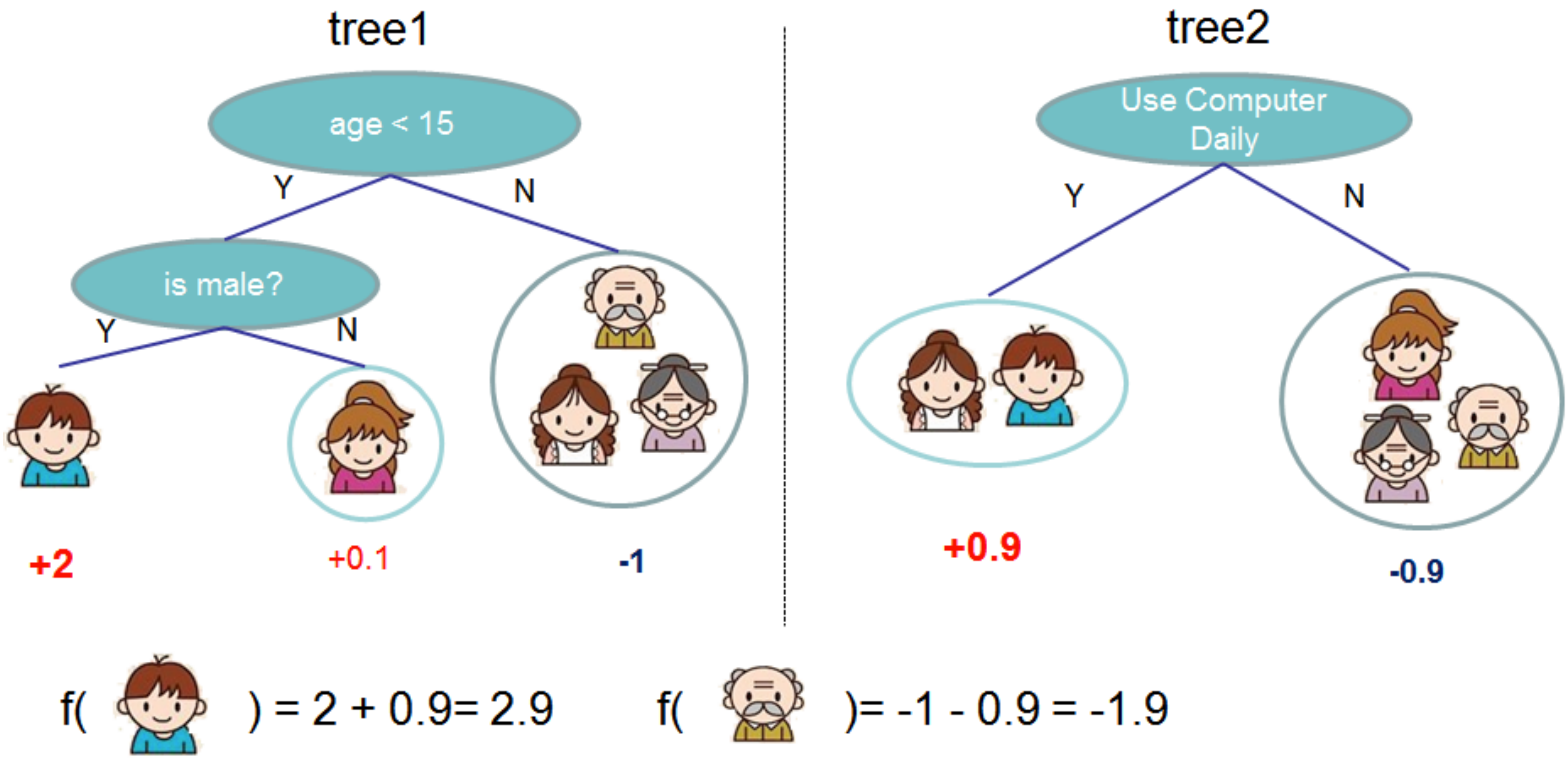
https://github.com/dataworkshop/model_evaluation/blob/master/step1-regression.ipynb

Decision Tree



http://xgboost.readthedocs.io/en/latest/model.html

Ensemble trees



http://xgboost.readthedocs.io/en/latest/model.html

Ensemble

- **Bagging** (bootstrap aggregation)
  - Random Forest
  - Extra Trees
- **Boosting**
  - Gradient Boosting

XGBoost

(eXtreme Gradient Boosting)

"When in doubt, use xgboost"

Owen Zhang

OWEN'S PROFILE ON KAGGLE

Model Selection

```
models = [
    ('extra_tree', ExtraTreesRegressor()),
    ('random_forest', RandomForestRegressor()),
    ('bagging', BaggingRegressor()),
    ('ada boost', AdaBoostRegressor()),
    ('gradient boosting', GradientBoostingRegressor()),
    ('decision tree', DecisionTreeRegressor())
]


for model_name, model in models:
    results = find_the_best_features(train, lambda k: SelectKBest(f_regression, k), model=model, v
    the_best_features = results[0][2]

    mean_score, std_score = count_modeling(model, train, the_best_features)
    print("model_name={0}: score={1}, features={2}".format(model_name, mean_score, the_best_featur
```

Understand Business & Data

Read and explore data

datetime	season	temp	count
2011-01-01 08:32:02	1	9.23	5
2012-04-02 12:10:00	2	18.78	32
2012-08-07 15:47:01	3	15.45	15

Feature Engineering

Create a new ones based on already exists

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Feature Selection

Select only useful features

~~datetime~~	season	temp	hour	day	month	~~...~~	count	count_log
~~2011-01-01 08:32:02~~	1	9.23	8	1	1	~~...~~	5	1.609
~~2012-04-02 12:10:00~~	2	18.78	12	2	4	~~...~~	32	3.466
~~2012-08-07 15:47:01~~	3	15.45	15	7	8	~~...~~	15	2.708

Model Selection

Find the best model(s)

model A | model B | model C | model D | model E

Tuning Hyperparameters

Find the best hyperparameters for given model

model B

model E

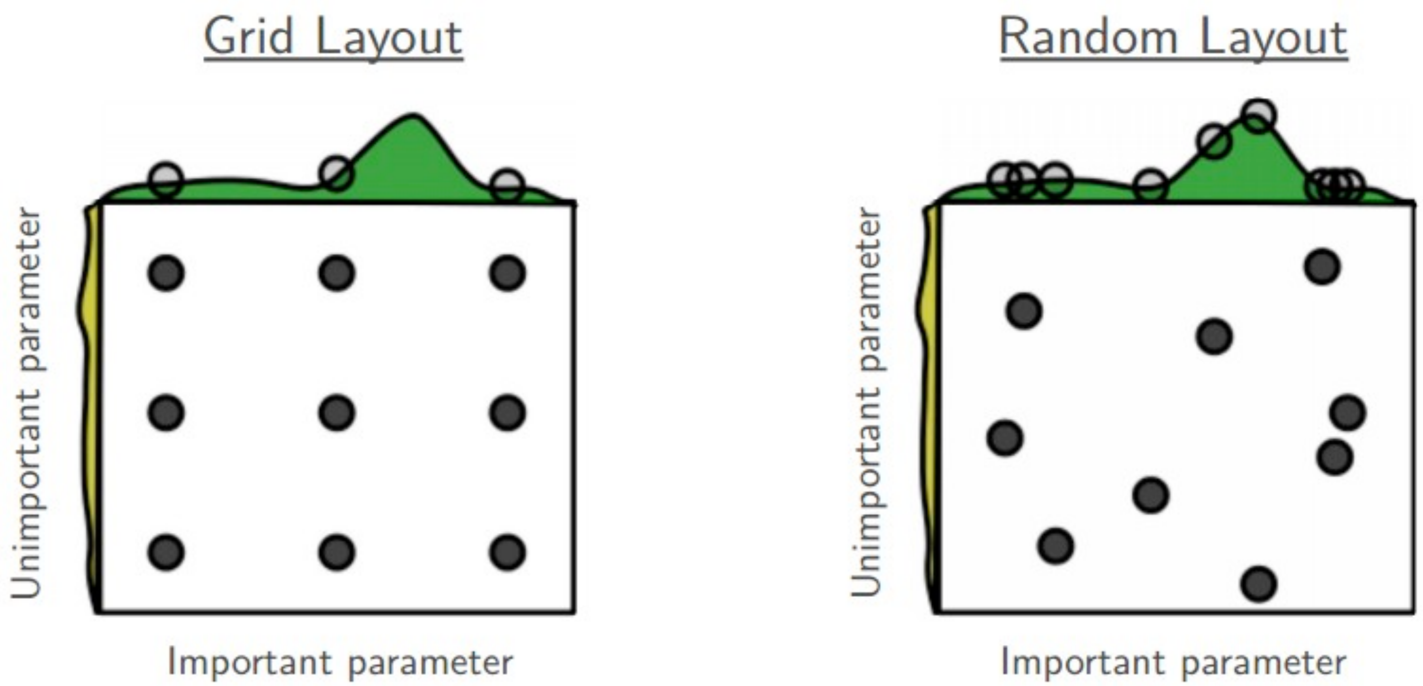
Ensemble Modeling

Combine few models into one more better

model B x0.6 + model E x0.4

Tuning Hyperparameters

- Grid Search
- Random Search
- **Bayesian**



hyperopt

```
def objective(space):

    model = xgb.XGBRegressor(
        max_depth = space['max_depth'],
        n_estimators = int(space['n_estimators']))


    X_train, X_test, y_train, y_test = train_test_split(train, 'count')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    score = rmsle(y_test, y_pred)

    return{'loss':score, 'status': STATUS_OK }

space ={
    'max_depth': hp.quniform("x_max_depth", 2, 20, 1),
    'n_estimators': hp.quniform("n_estimators", 100, 1000, 1),
}


trials = Trials()
best = fmin(fn=objective,
            space=space,
            algo=tpe.suggest,
            max_evals=15,
            trials=trials)
```

Understand Business & Data

Read and explore data

datetime	season	temp	count
2011-01-01 08:32:02	1	9.23	5
2012-04-02 12:10:00	2	18.78	32
2012-08-07 15:47:01	3	15.45	15

Feature Engineering

Create a new ones based on already exists

datetime	season	temp	hour	day	month	...	count	count_log
2011-01-01 08:32:02	1	9.23	8	1	1	...	5	1.609
2012-04-02 12:10:00	2	18.78	12	2	4	...	32	3.466
2012-08-07 15:47:01	3	15.45	15	7	8	...	15	2.708

Feature Selection

Select only useful features

~~datetime~~	season	temp	hour	day	month	~~...~~	count	count_log
~~2011-01-01 08:32:02~~	1	9.23	8	1	1	~~...~~	5	1.609
~~2012-04-02 12:10:00~~	2	18.78	12	2	4	~~...~~	32	3.466
~~2012-08-07 15:47:01~~	3	15.45	15	7	8	~~...~~	15	2.708

Model Selection

Find the best model(s)

model A | model B | model C | model D | model E

Tuning Hyperparameters

Find the best hyperparameters for given model

model B | model E

Ensemble Modeling

Combine few models into one more better

model B **x0.6** + model E **x0.4**

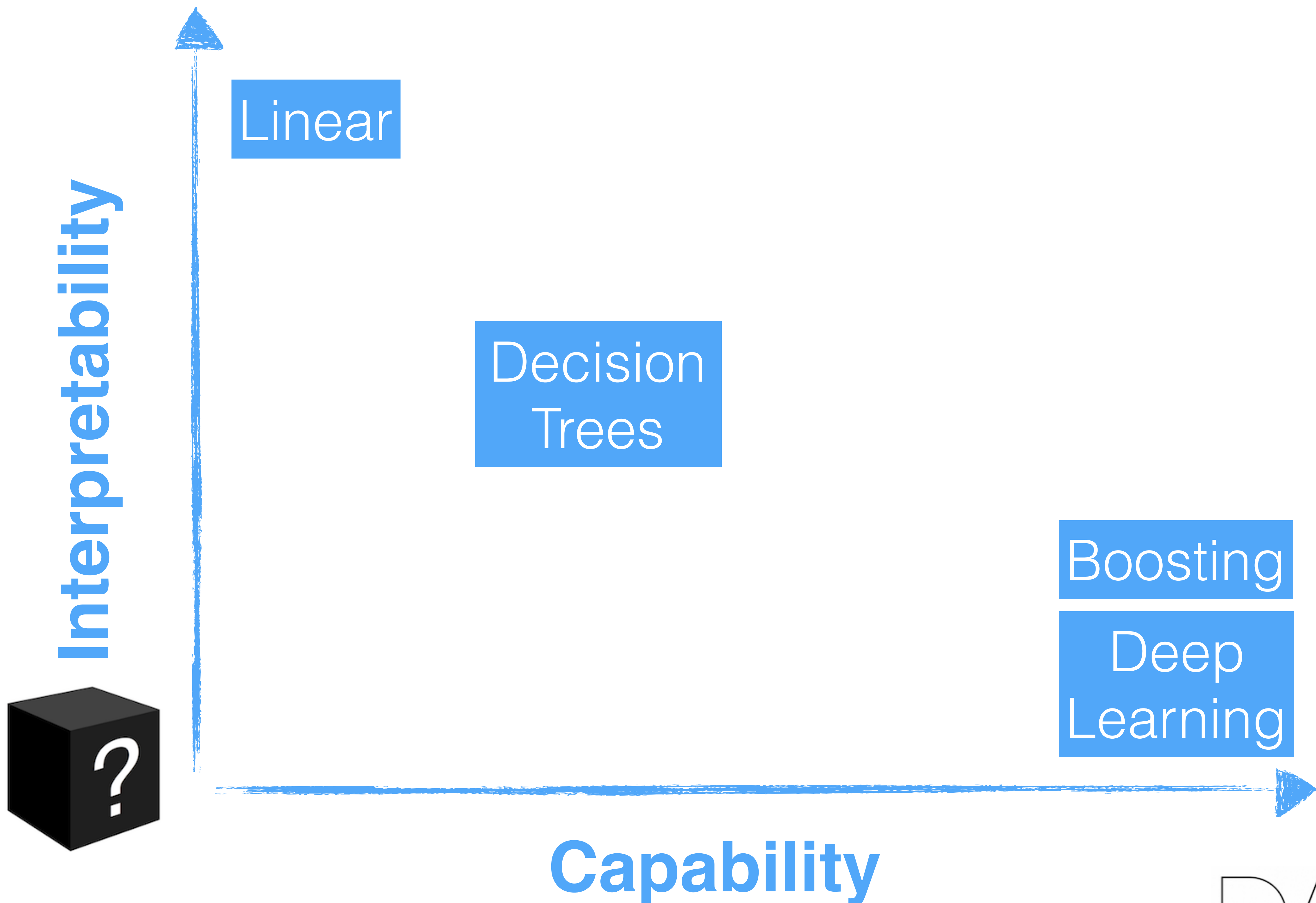
Ensemble Modeling

```
gbm_params = {'n_estimators': 150, 'max_depth': 5, 'random_state': 0, 'min_samples_
rf_params = {'n_estimators': 1000, 'max_depth': 15, 'random_state': 0, 'min_samples
xgb_params = {'n_estimators':150,  'learning_rate':0.1, 'max_depth':5, 'subsample':


gbm_count = predict(gbm_X_train, gbm_X_test, GradientBoostingRegressor(**gbm_params
rf_count = predict(rf_X_train, rf_X_test, RandomForestRegressor(**rf_params))
xgb_count = predict(xgb_X_train, xgb_X_test, xgb.XGBRegressor(**xgb_params))

test['count'] = .2*rf_count + .8*( .3*gbm_count  + .7*xgb_count)
```

Deep Learning :)



DATA WORKSHOP

DeepMind
@DeepMindAI

Follow

Using cognitive psychology to address 'black box' interpretability. Read our blog at deepmind.com/blog/cognitive… by S. Ritter & @dgtbarrett

MIT Technology Review

Intelligent Machines

Nvidia Lets You Peer Inside the Black Box of Its Self-Driving AI

In a step toward making AI more accountable, Nvidia has developed a neural network for autonomous driving that highlights what it's focusing on.

by Will Knight May 3, 2017

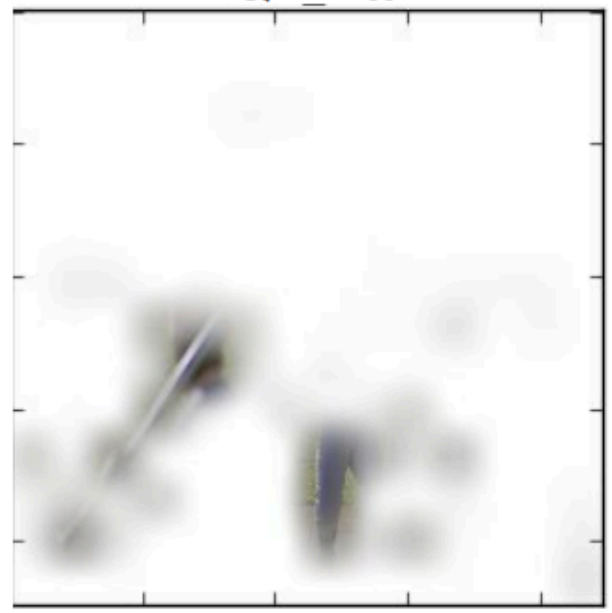
DATA WORKSHOP

Q: What sport is this?

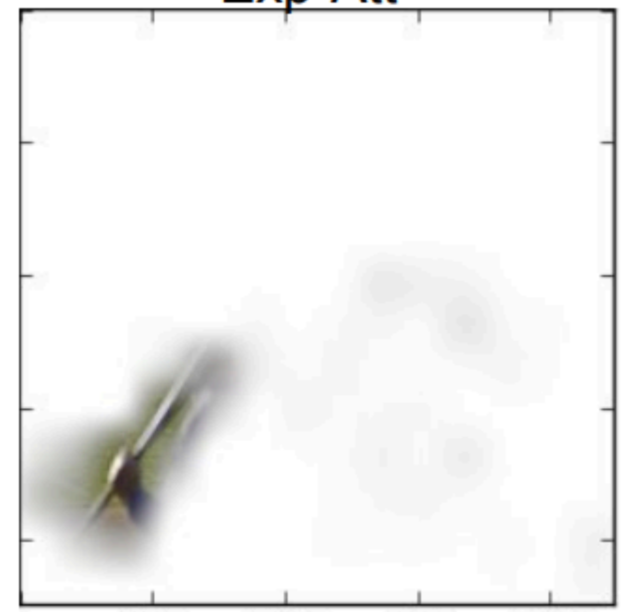
A: Baseball

Textual Justification:
The player is holding a bat

<VQA-Att>



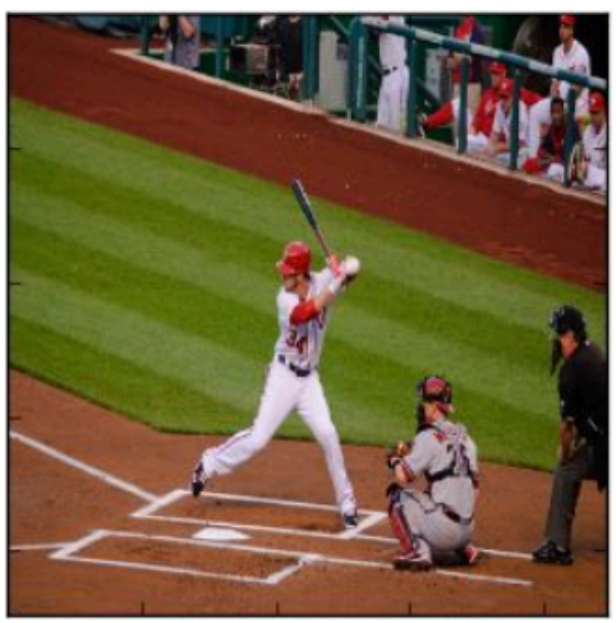
<Exp-Att>



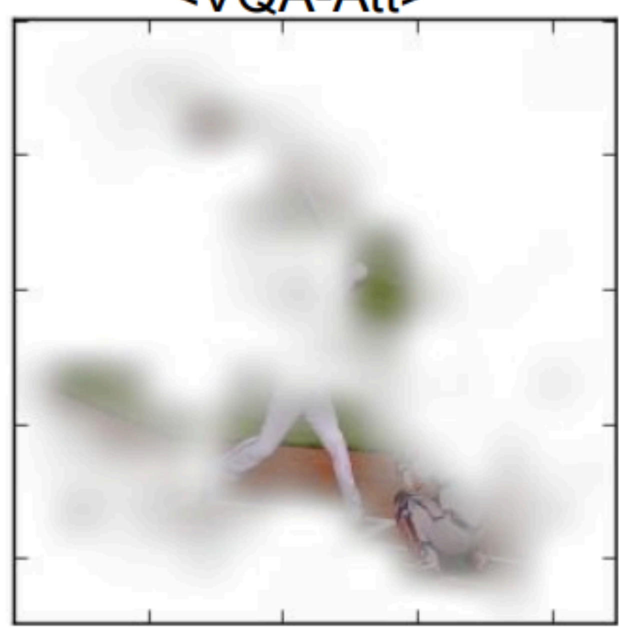
Q: What sport is this?

A: Baseball

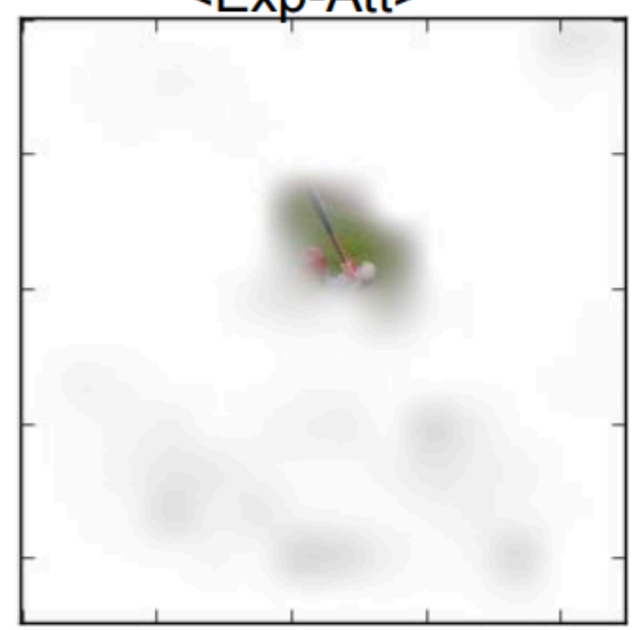
Textual Justification:
The player is swinging a bat



<VQA-Att>



<Exp-Att>



why
***electric guitar*?**

why
***acoustic guitar*?**

why
***labrador*?**

original image

https://arxiv.org/pdf/1602.04938.pdf

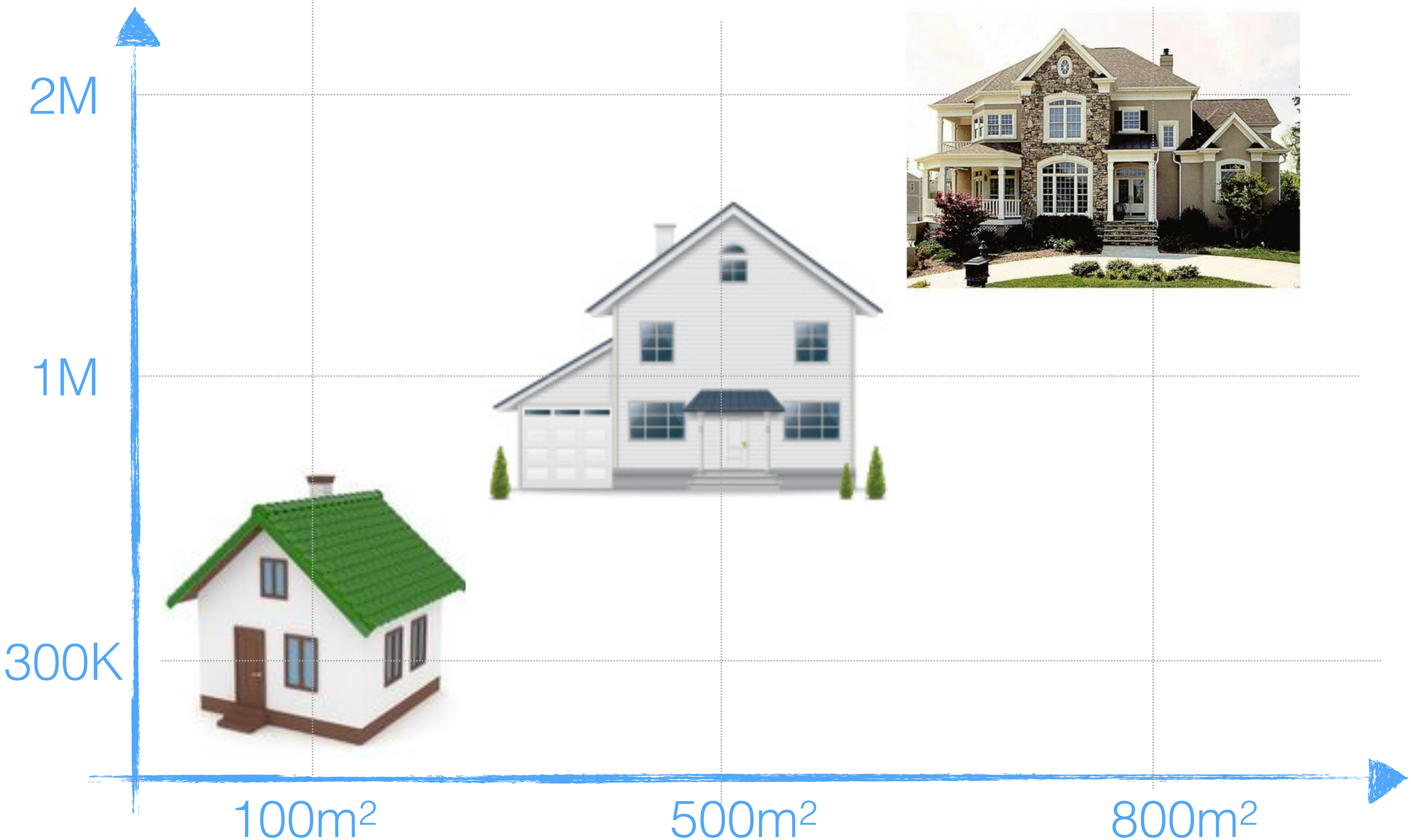
DATA WORKSHOP

Husky classified as Wolf

https://arxiv.org/pdf/1602.04938.pdf

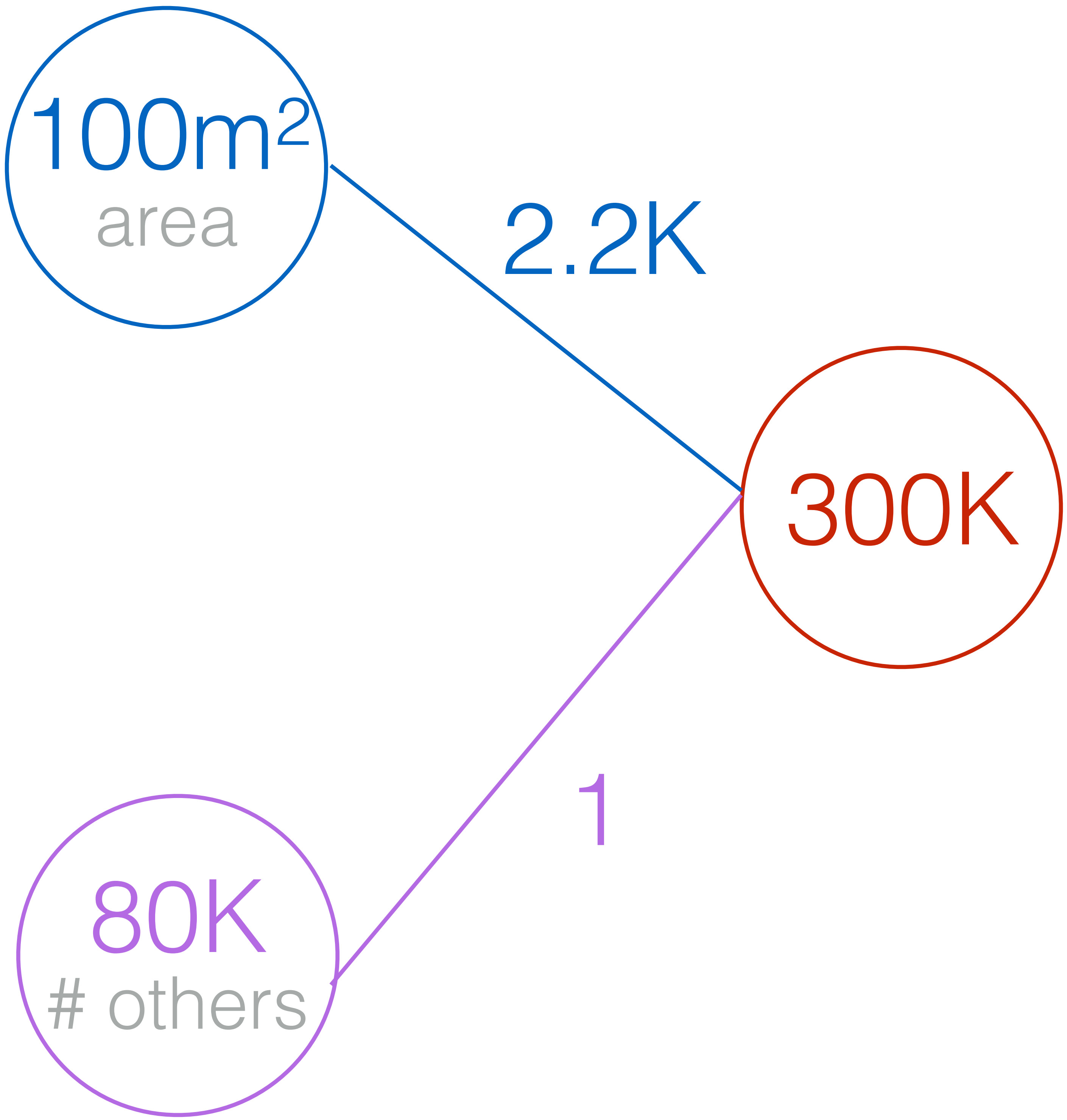
DATA WORKSHOP

Price of House

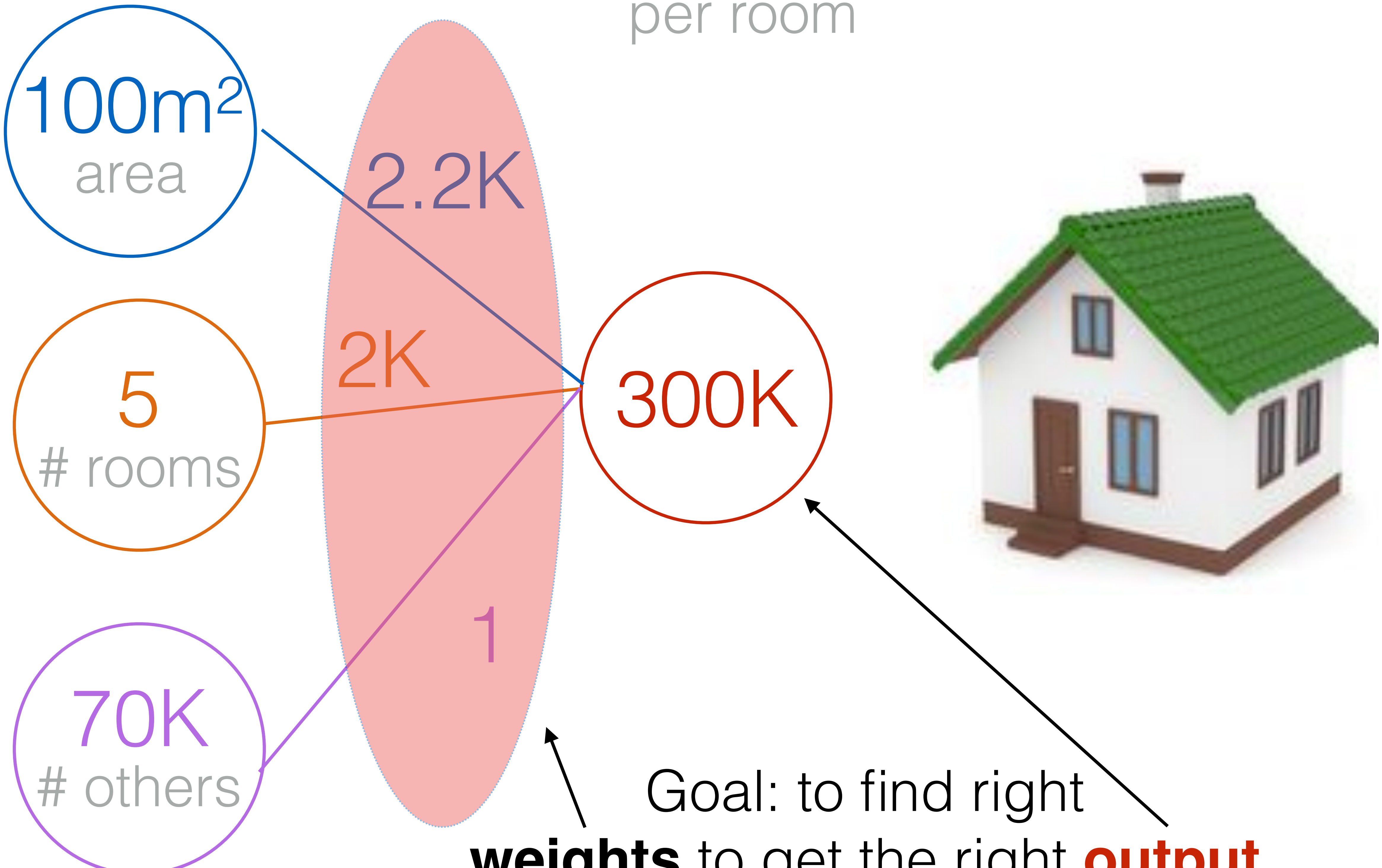


$100m^2$ x $2.2K$ + $80K$ = $300K$

area | cost m^2 | others

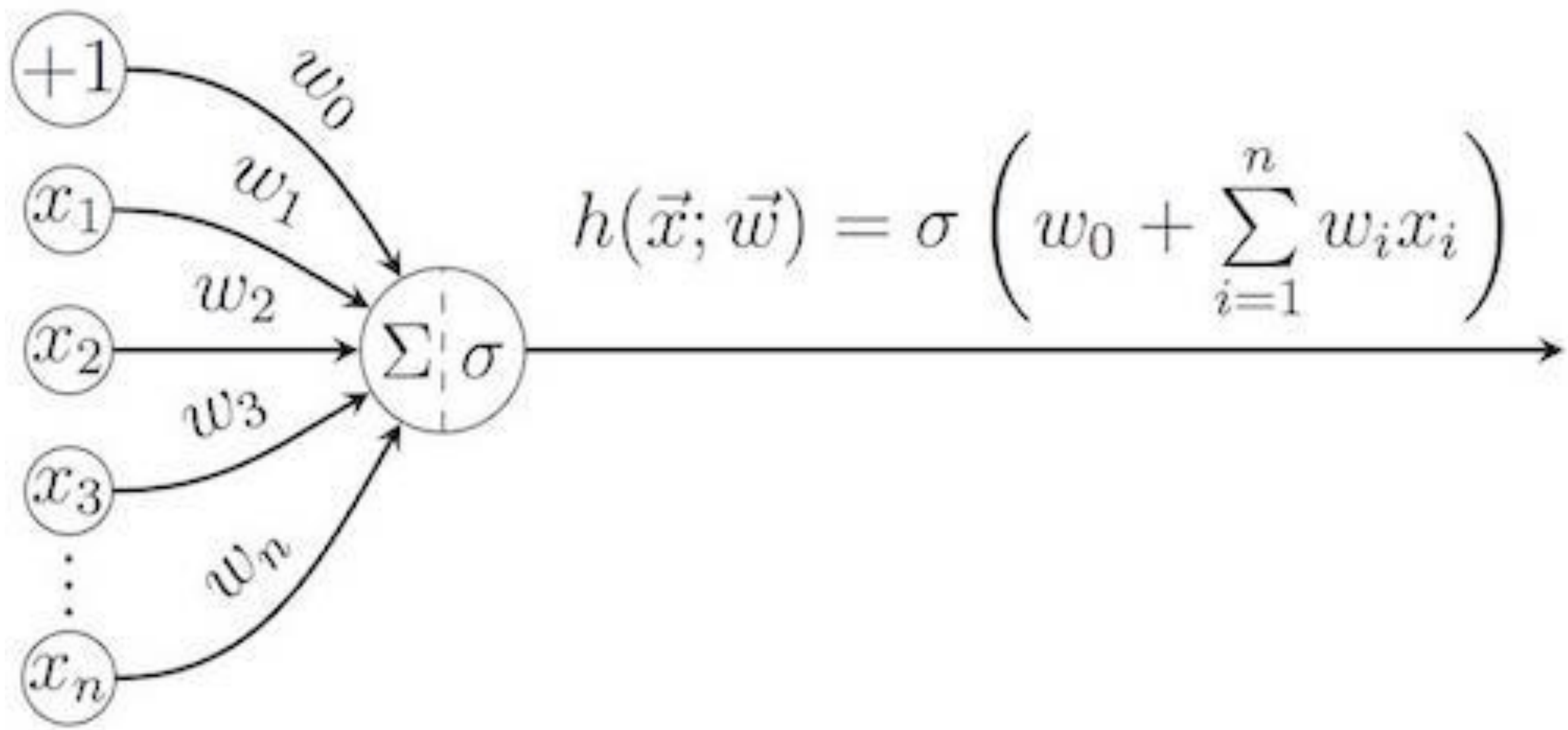


$$\underbrace{100m^2}_{\text{area}} \times \underbrace{2.2K}_{\text{cost } m^2} + \overbrace{5}^{\#\text{rooms}} \times \underbrace{2K}_{\text{cost per room}} + \underbrace{70K}_{\text{others}} = 300K$$



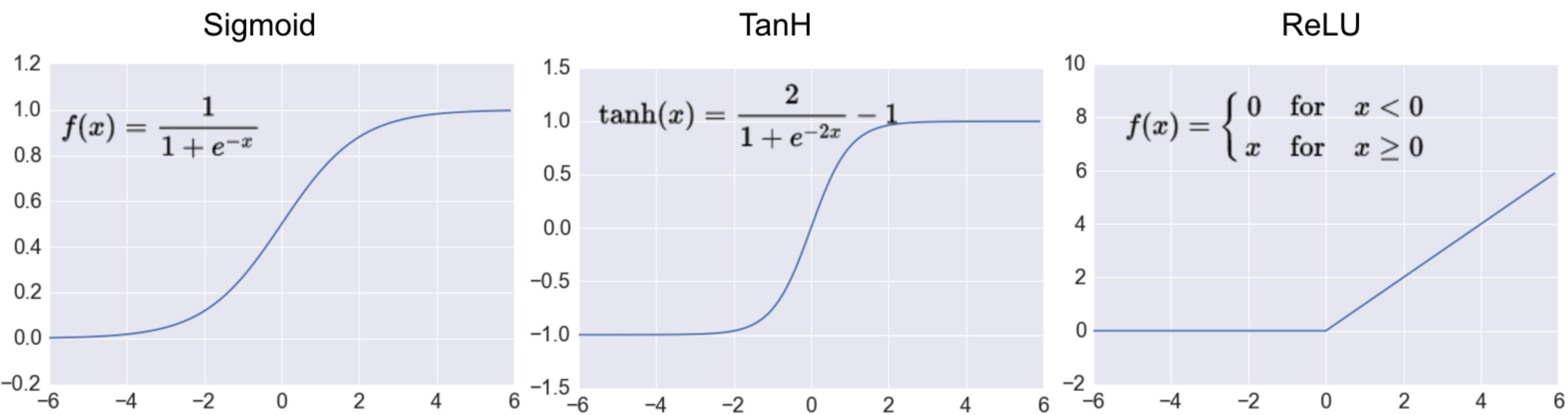
Goal: to find right **weights** to get the right **output**

Neuron



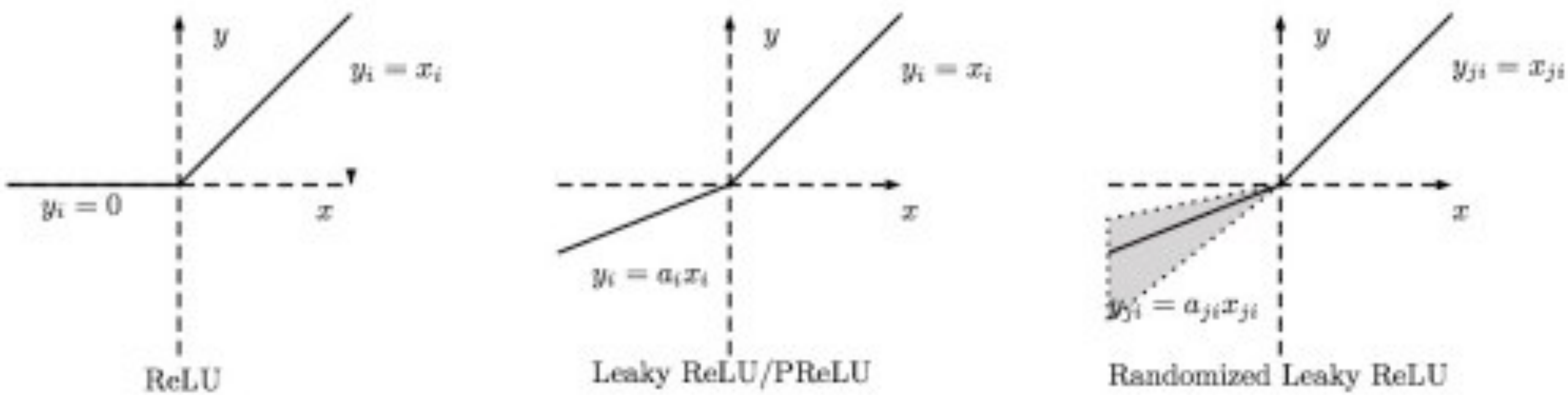
$+1$
x_1
x_2
x_3
x_n
w_0
w_1
w_2
w_3
w_n
Σ
σ
$h(\vec{x};\vec{w}) = \sigma\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$

Activation function



- Sigmoid , Tanh , ReLu
- PReLu
- SoftMax (normalized exponential function)

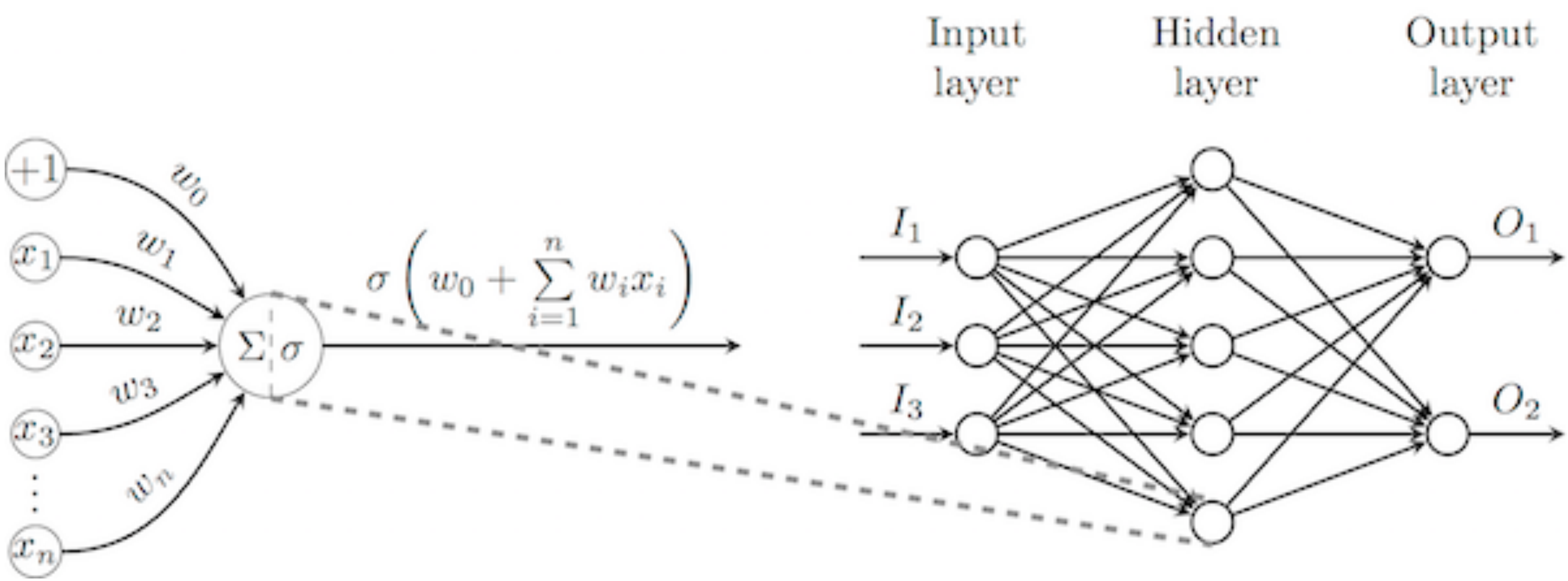
Parametric ReLU at. el.



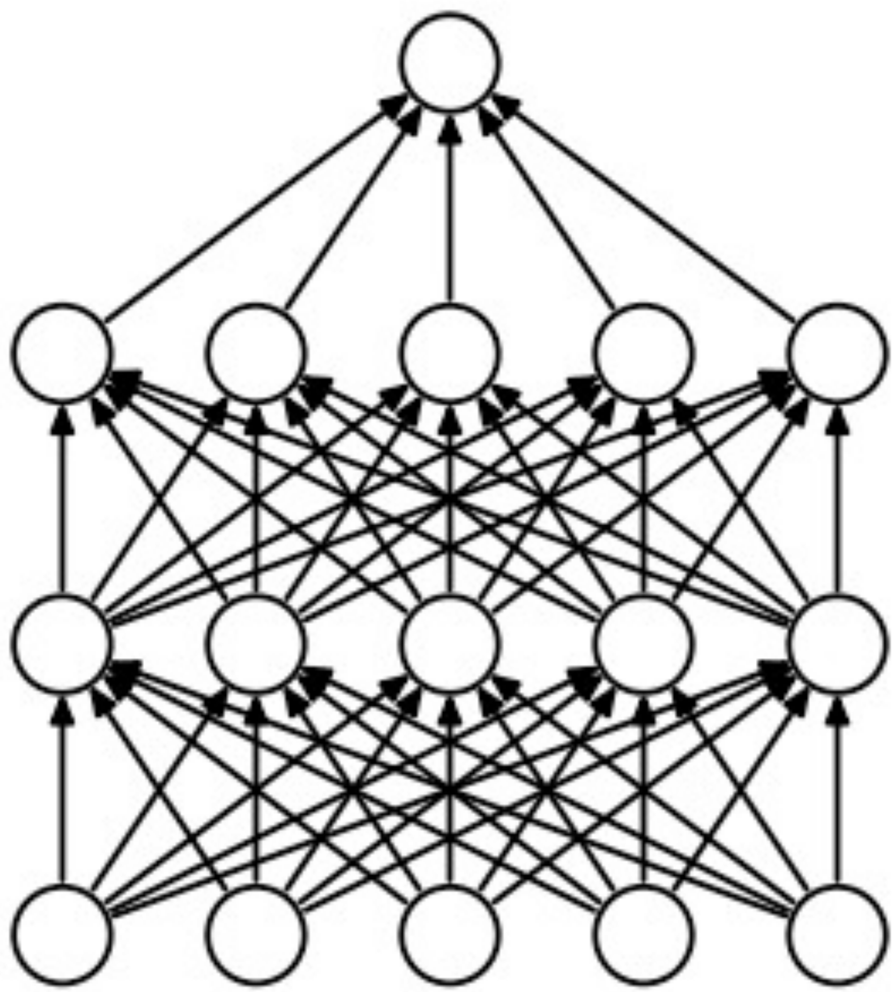
Activation	Training Error	Test Error
ReLU	0.00318	0.1245
Leaky ReLU, $a = 100$	0.0031	0.1266
Leaky ReLU, $a = 5.5$	0.00362	**0.1120**
PReLU	0.00178	0.1179
RReLU	0.00550	**0.1119**

Table 3. Error rate of CIFAR-10 Network in Network with different activation function

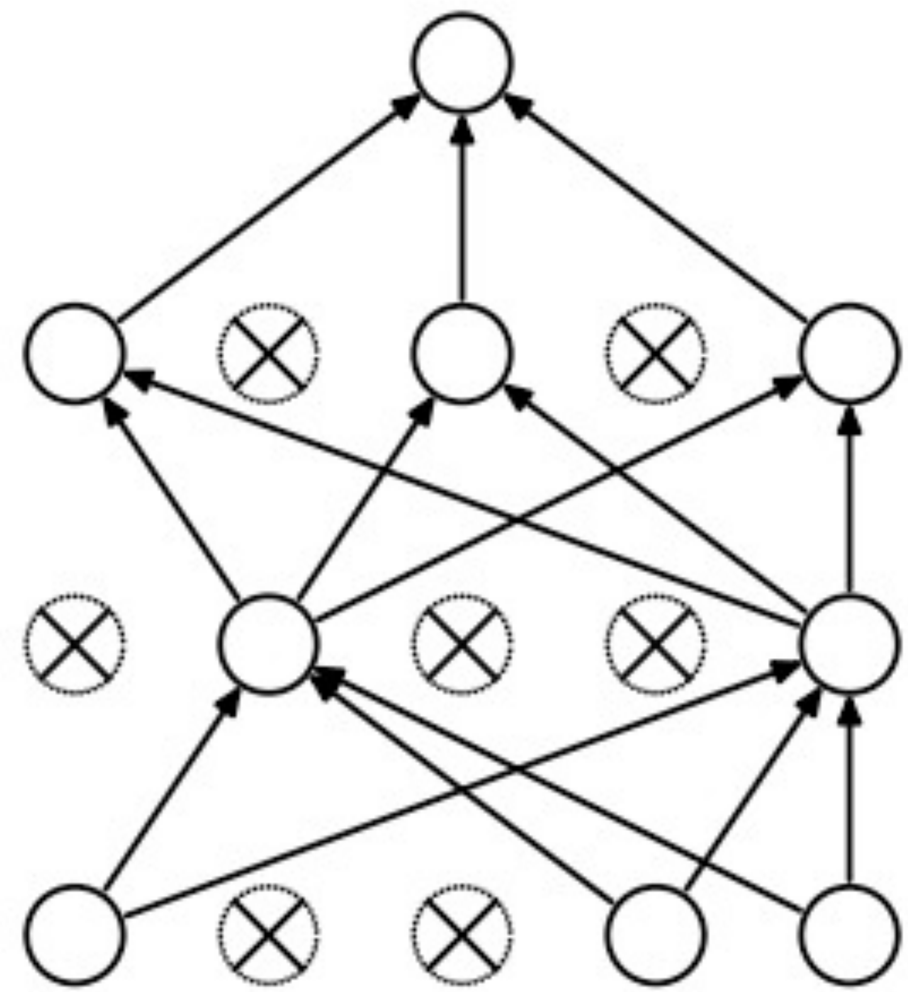
(Artificial) Neural Network



Dropout



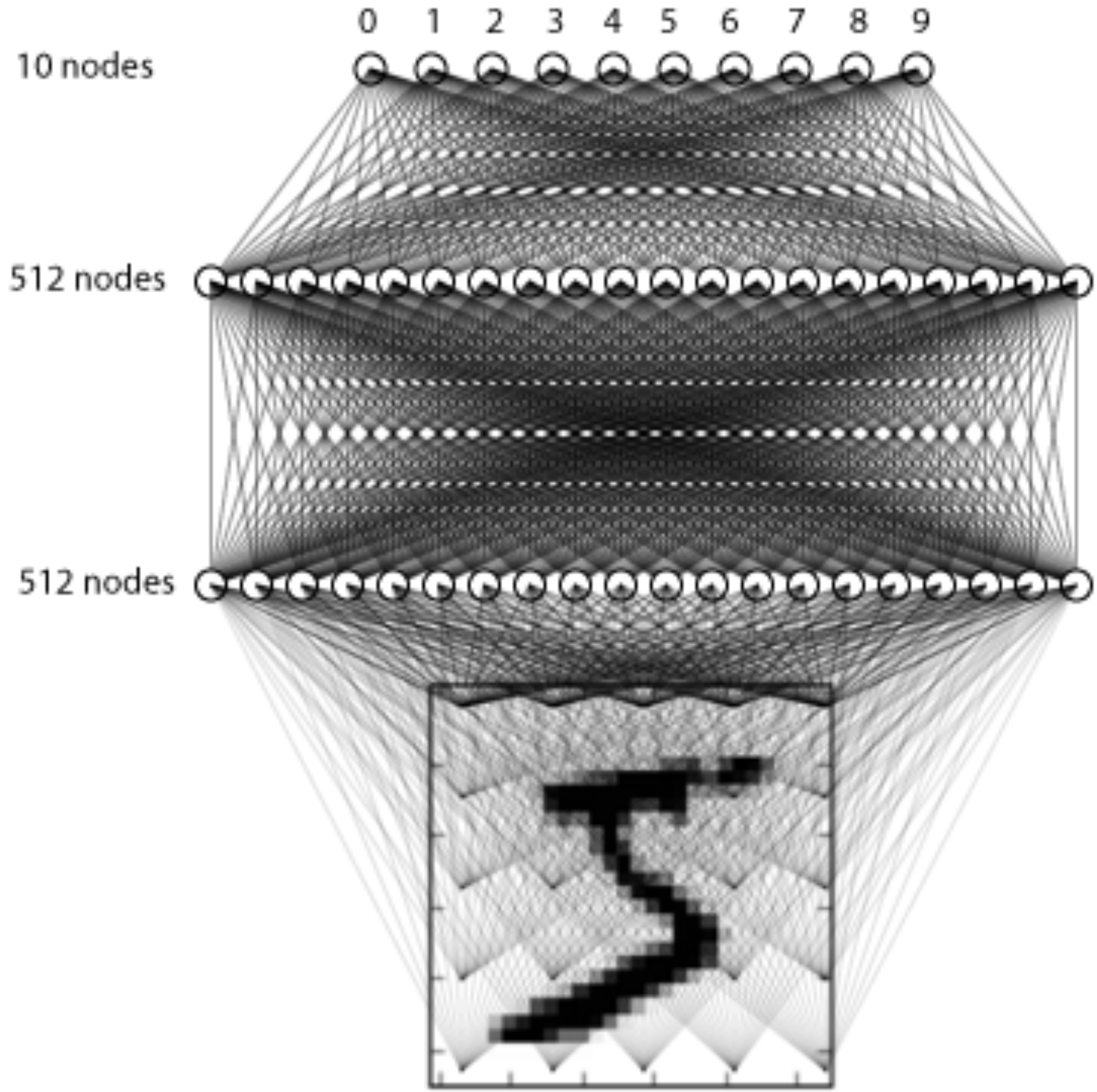
(a) Standard Neural Net



(b) After applying dropout.

http://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf

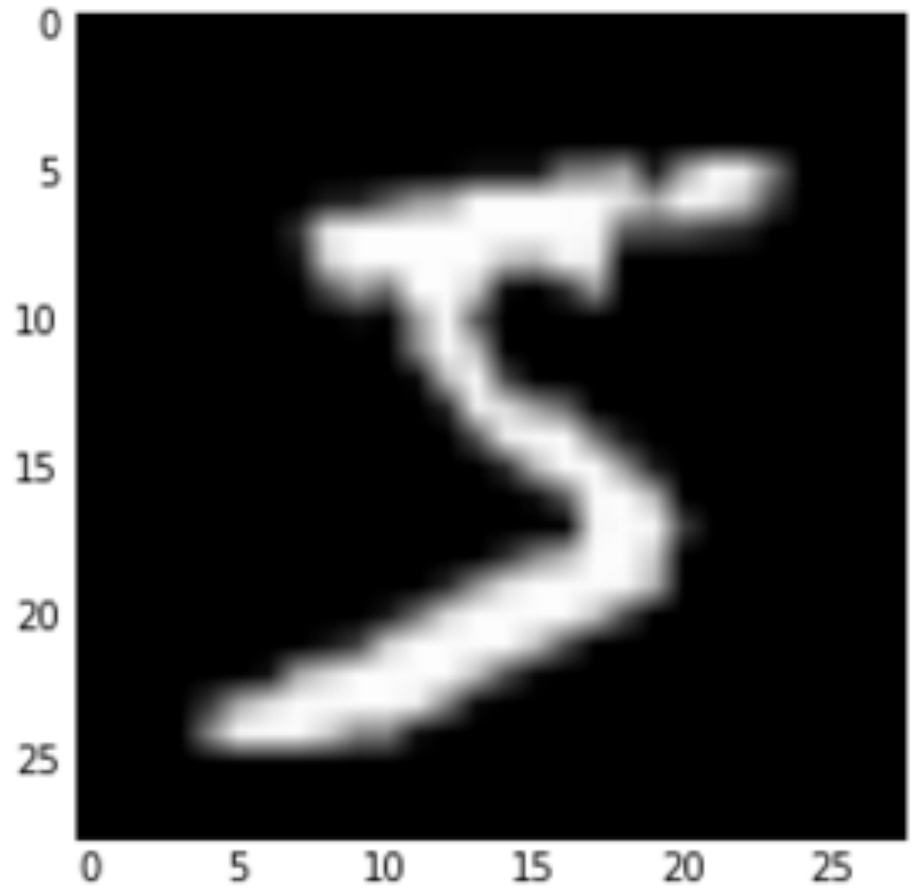
MNIST


0 1 2 3 4 5 6 7 8 9
10 nodes
512 nodes
512 nodes

Load Data

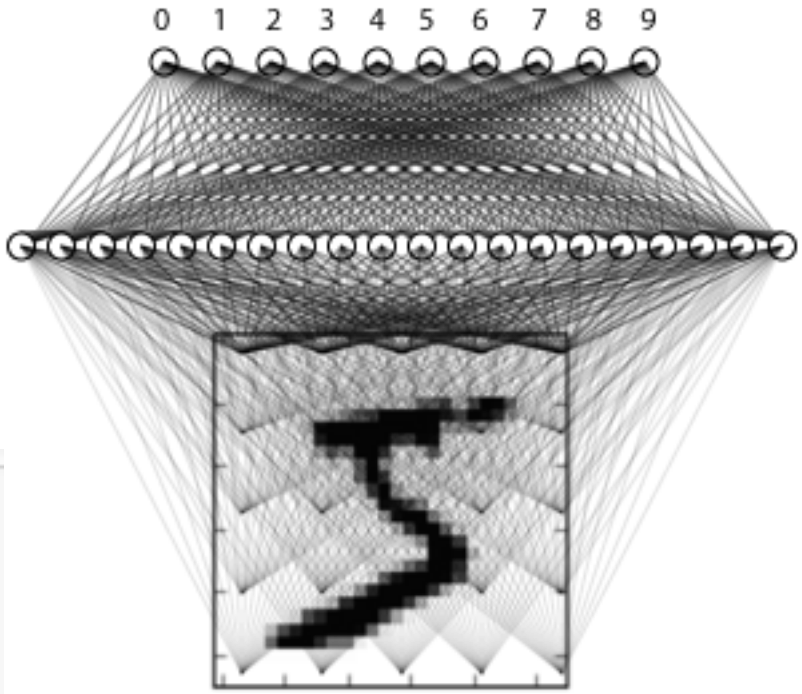
```
from keras.datasets import mnist

(X_train, y_train), (X_test, y_test) = mnist.load_data()
plt.imshow(X_train[0], cmap=plt.get_cmap('gray'))
plt.show()
```



Neural Network

Error: 1.60%




```
seed = 2016
numpy.random.seed(seed)

(X_train, y_train), (X_test, y_test) = mnist.load_data()

num_pixels = X_train.shape[1] * X_train.shape[2]
X_train = X_train.reshape(X_train.shape[0], num_pixels).astype('float32')
X_test = X_test.reshape(X_test.shape[0], num_pixels).astype('float32')

X_train = X_train / 255
X_test = X_test / 255

y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

model = Sequential()
model.add(Dense(num_pixels, input_dim=num_pixels, init='normal', activation='relu'))
model.add(Dense(num_classes, init='normal', activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_train, y_train, validation_data=(X_test, y_test), nb_epoch=20, batch_size=128,

scores = model.evaluate(X_test, y_test, verbose=0)
print("Baseline Error: %.2f%%" % (100-scores[1]*100))
```

MNIST 50 results collected

Units: error %

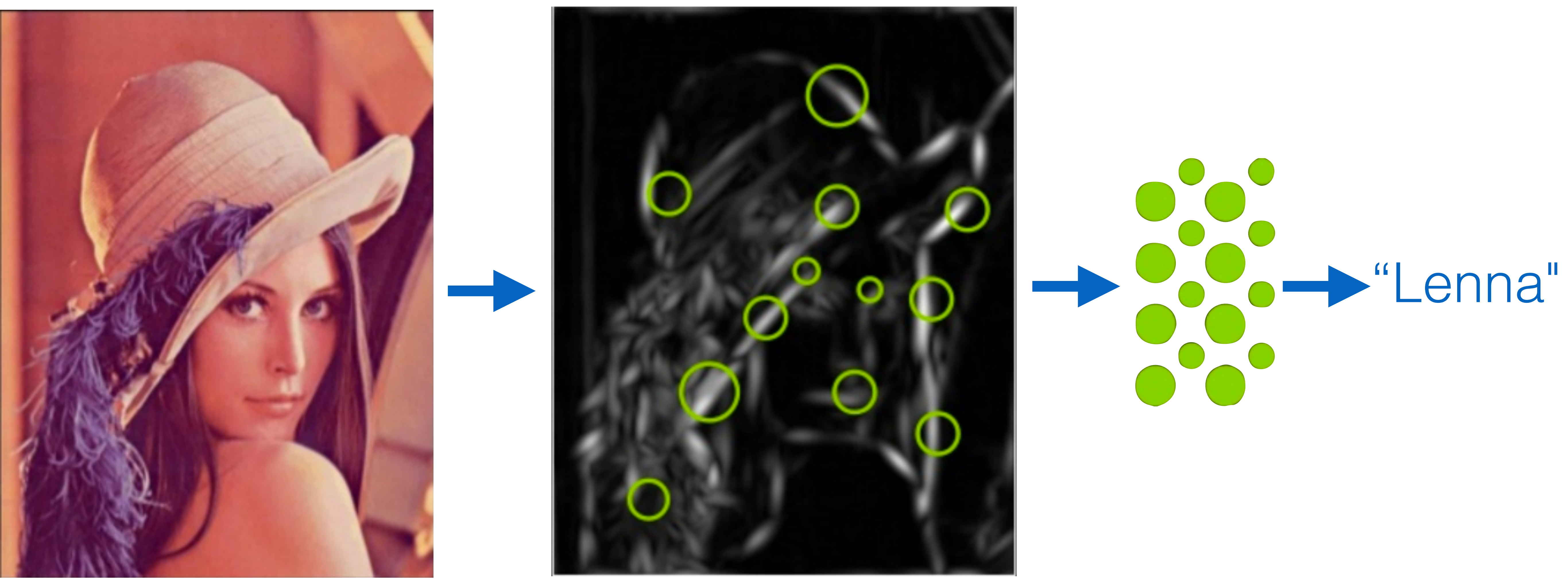
> Classify handwriten digits. Some additional results are available on the original dataset page.

Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
0.31%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	
0.31%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units	arXiv 2015	
0.32%	Fractional Max-Pooling	arXiv 2015	Details
0.33%	Competitive Multi-scale Convolution	arXiv 2015	
0.35%	Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition	Neural Computation 2010	Details
0.35%	C-SVDDNet: An Effective Single-Layer Network for Unsupervised Feature Learning	arXiv 2014	

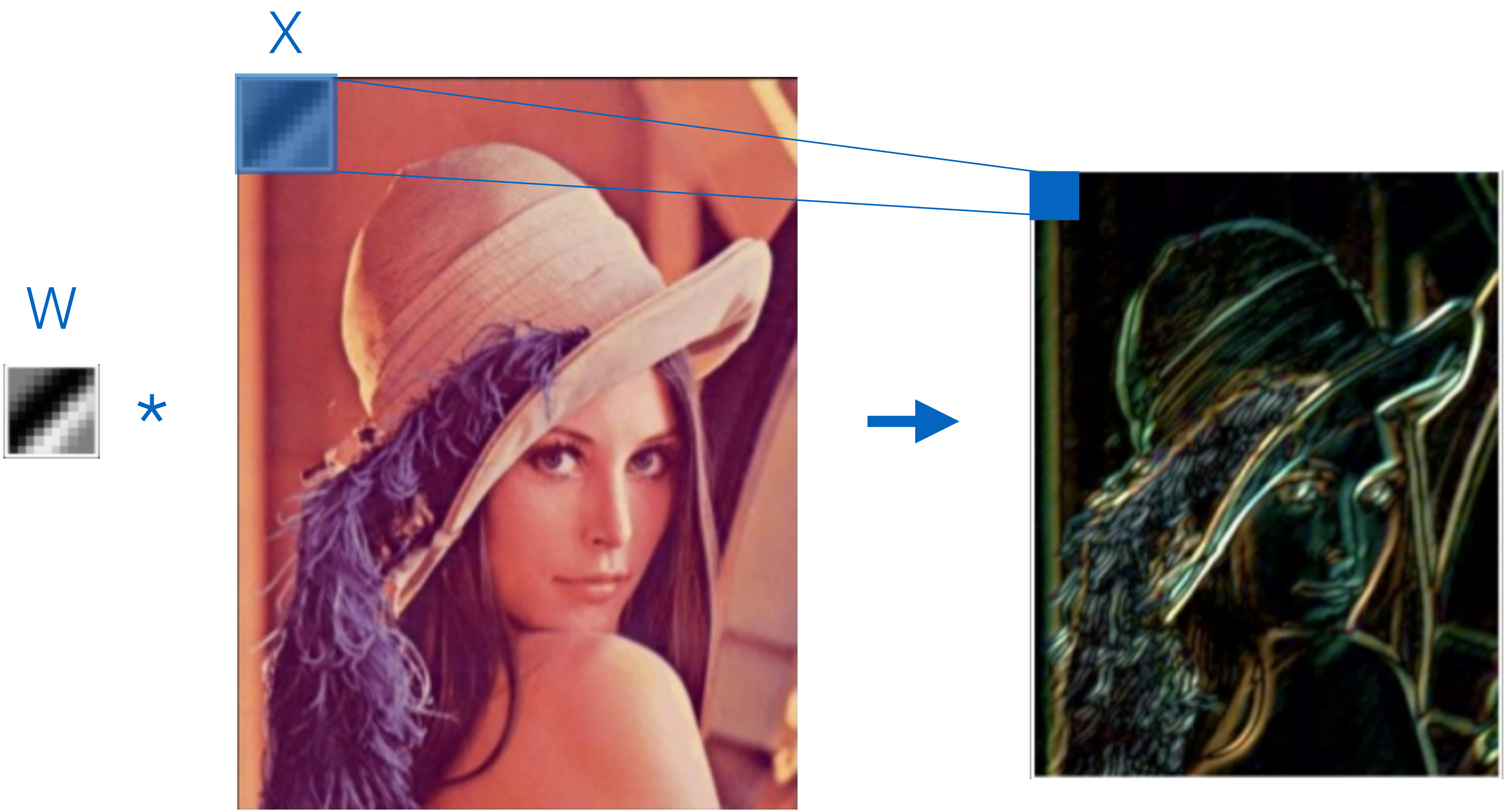
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

Computer Vision

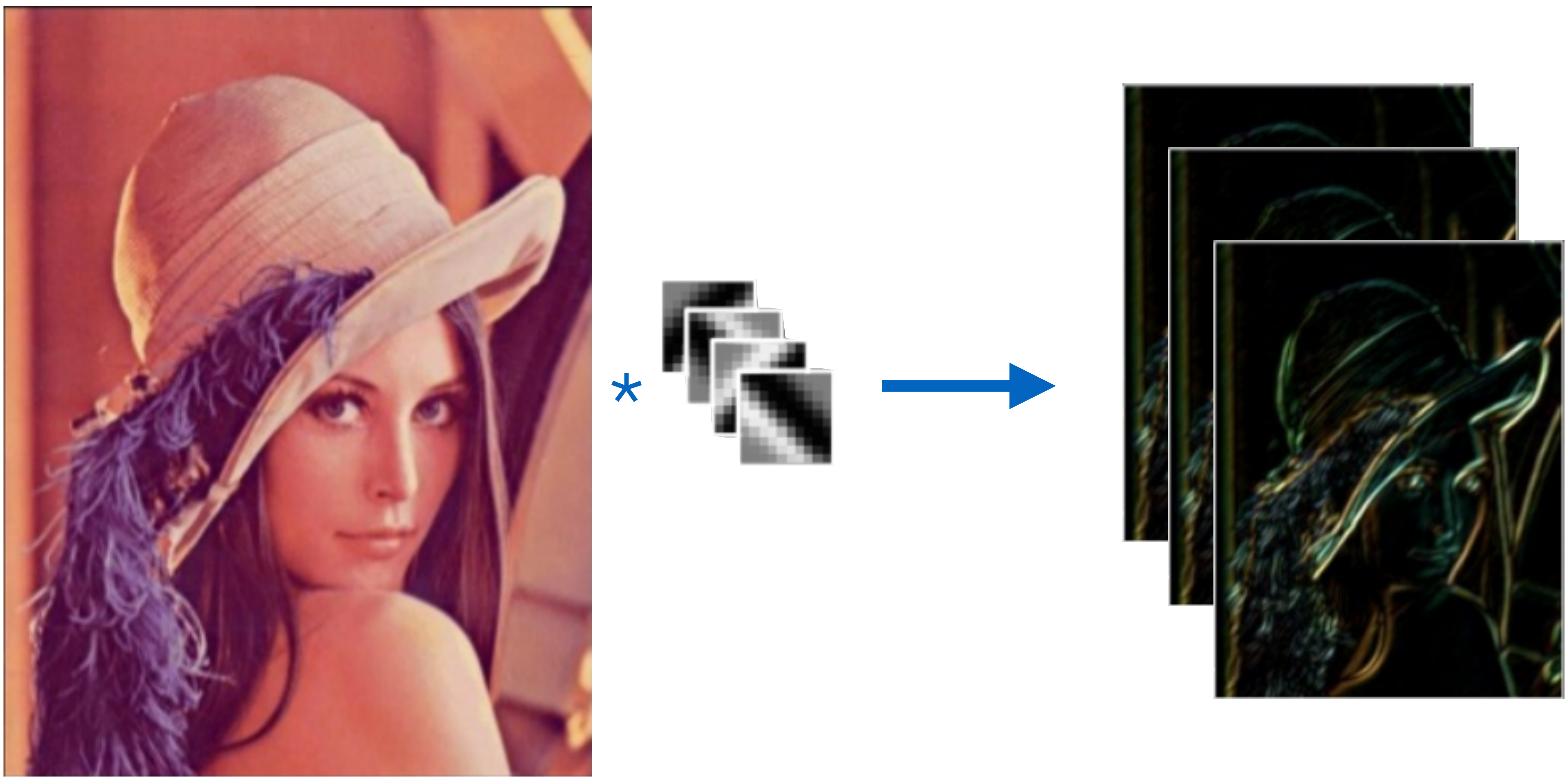
Original **Simplified** **Features**



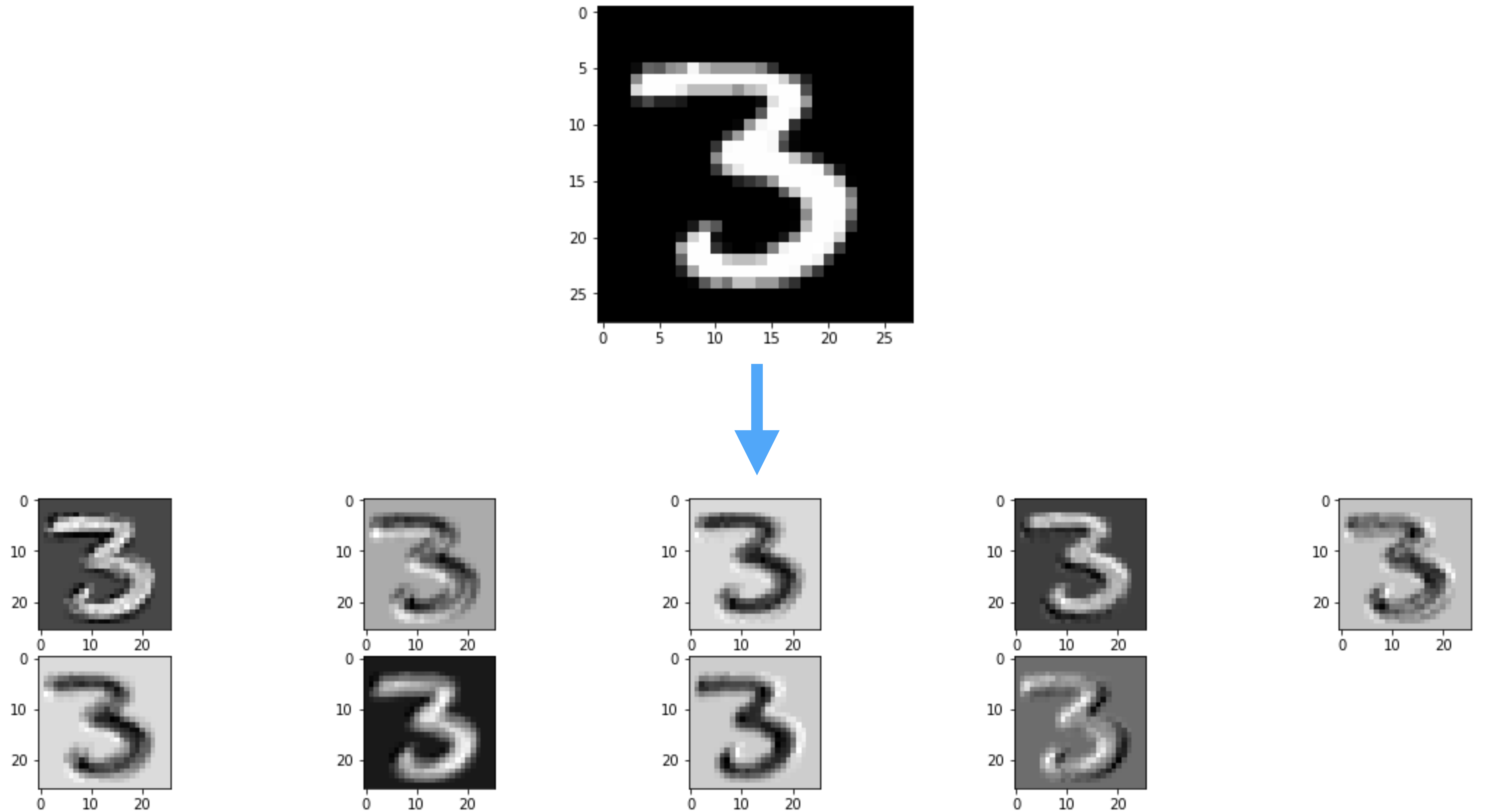
Feature Map



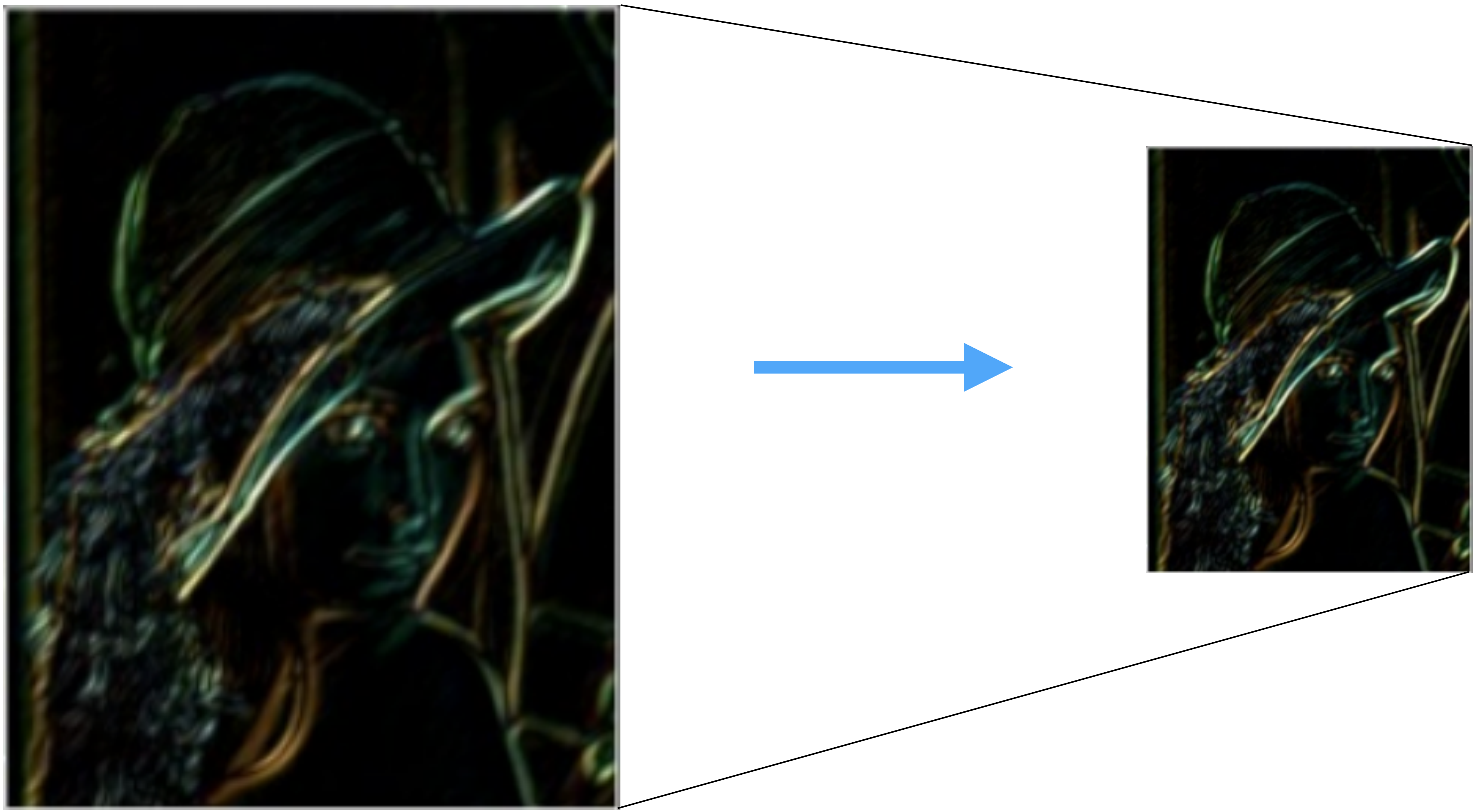
Feature Maps



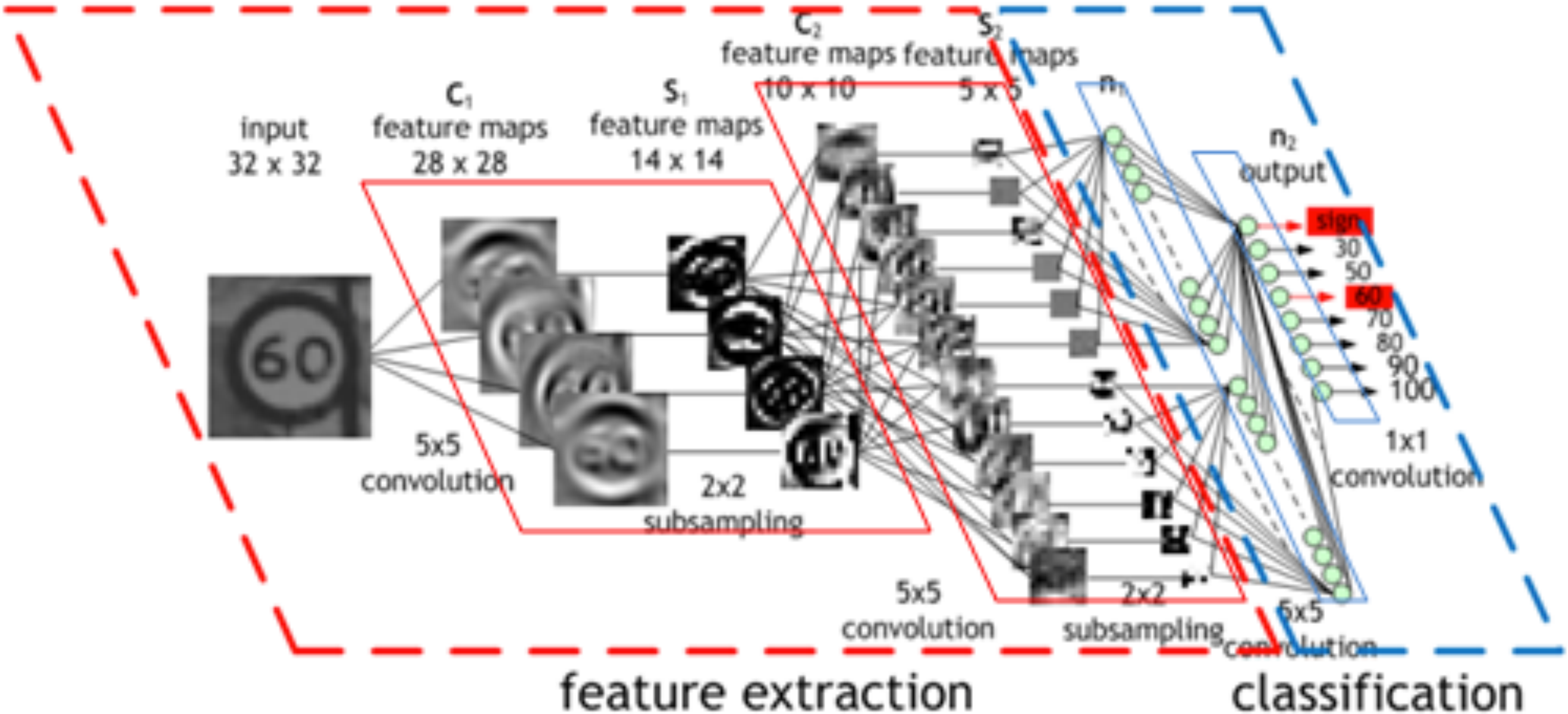
Feature Maps

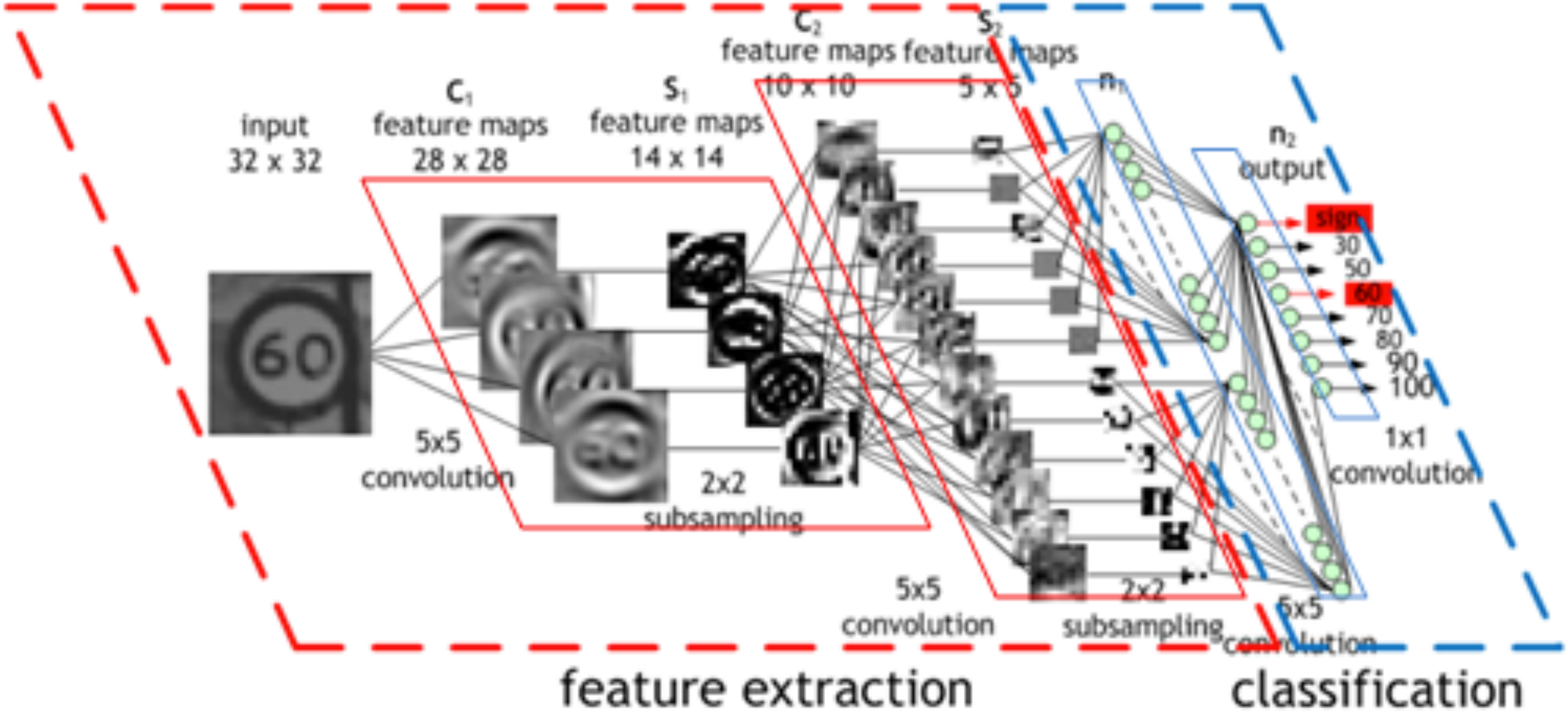


Pooling == resize



Convolutional Neural Networks



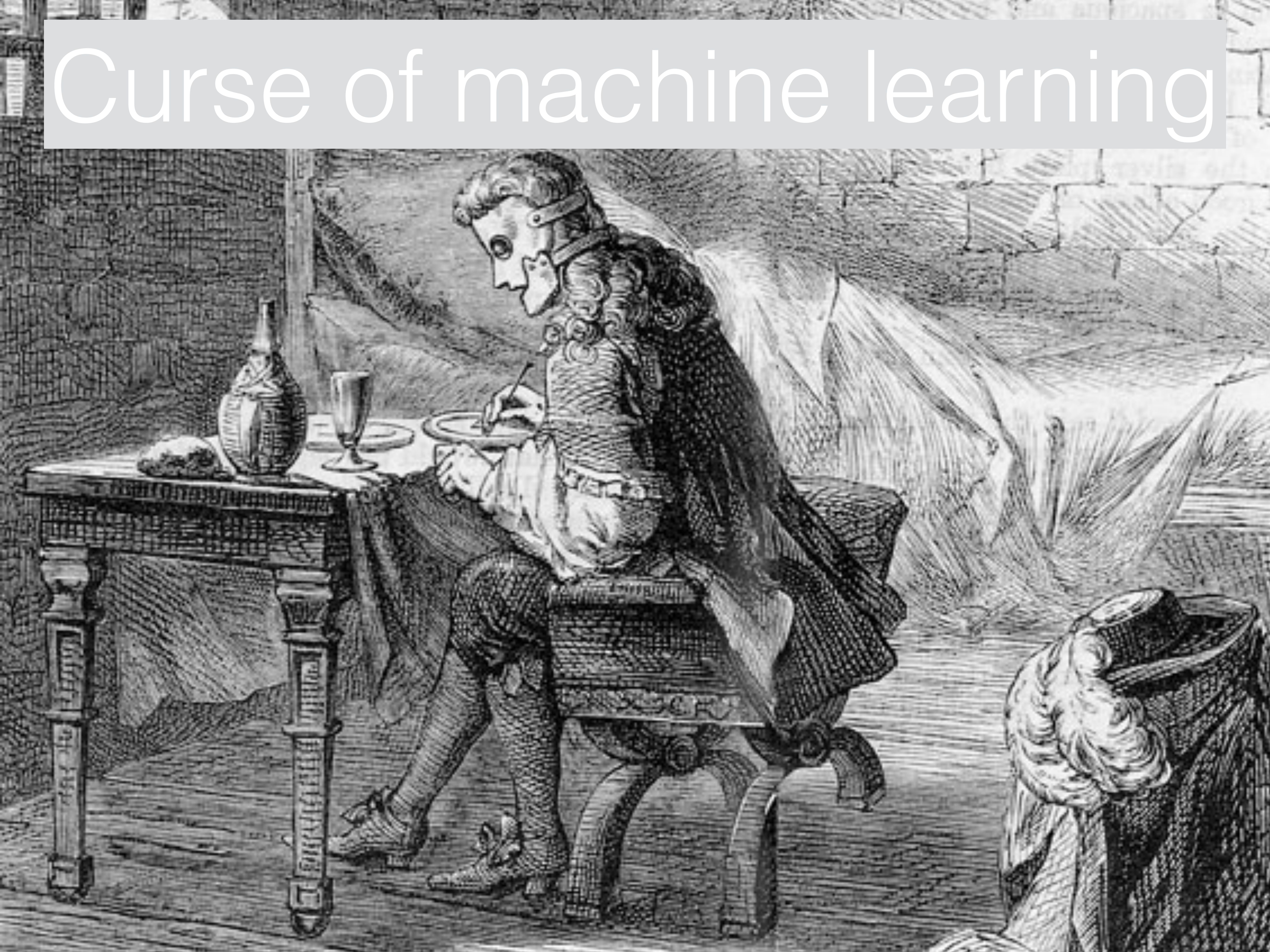



```
model = Sequential()

model.add(Conv2D(4, kernel_size=(5, 5), activation='relu', input_shape=(1, 32, 32)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(12, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(100, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])
```

Curse of machine learning



Reuse knowledge

Transfer Learning

Lifelong Learning

Gradual Learning

Curriculum Learning

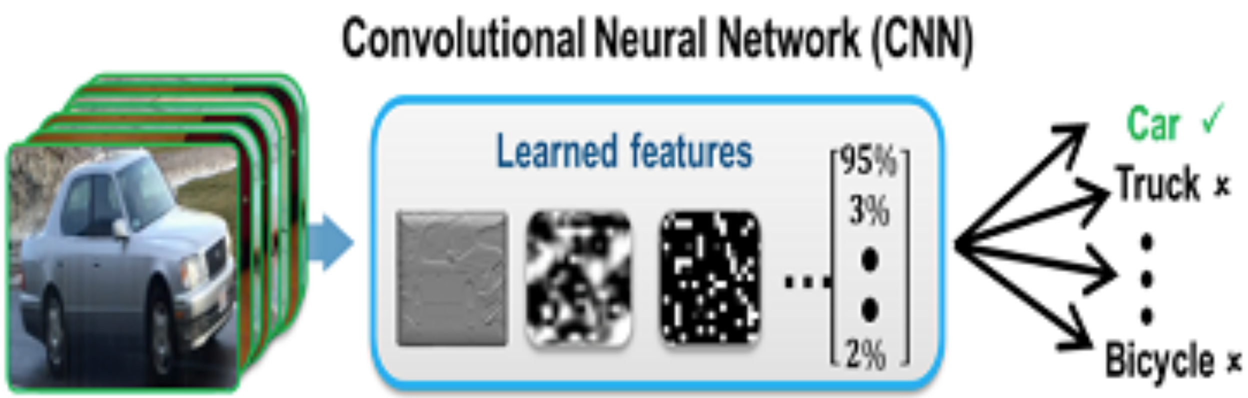
Compositional Learning

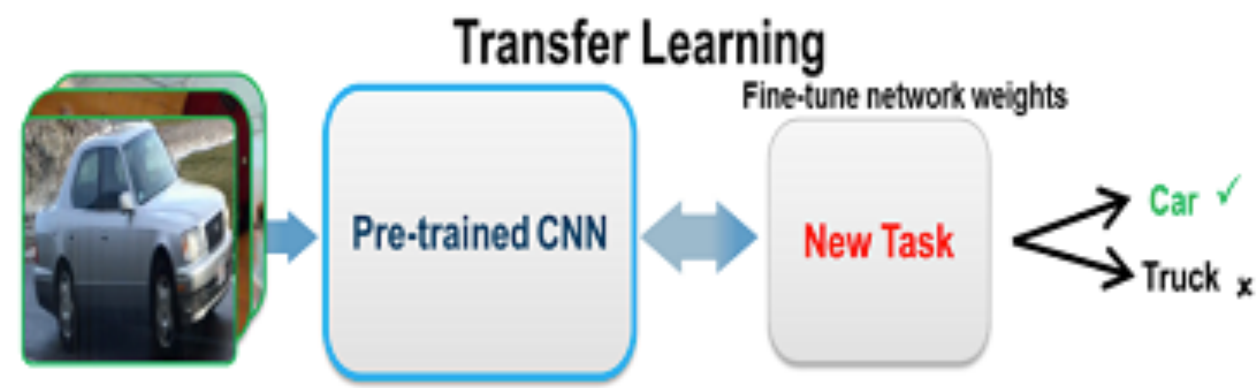
Continual Learning

Sequential Learning

Never-Ending Learning

Transfer Learning & Fine-tuning

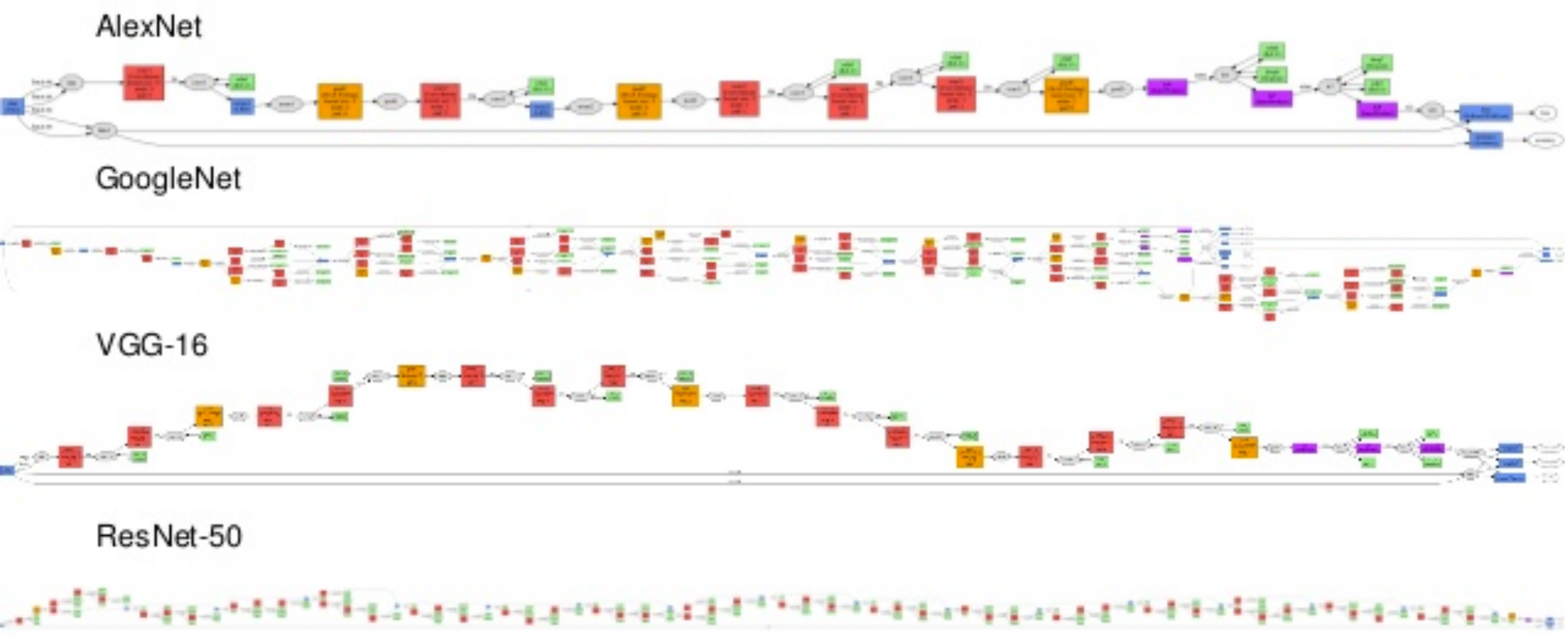




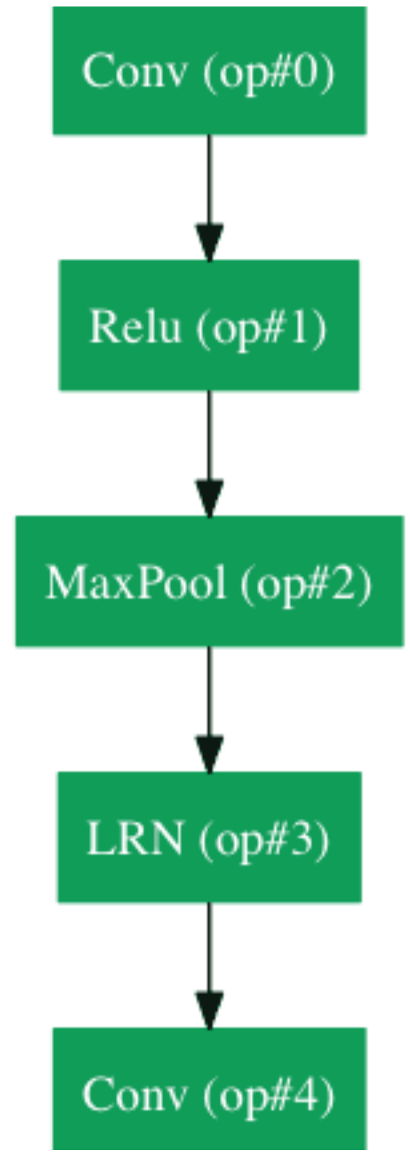
Training data	1000s to millions of labeled images
Computation	Compute intensive
Training Time	Days to Weeks for real problems
Model accuracy	High (can over fit to small datasets)

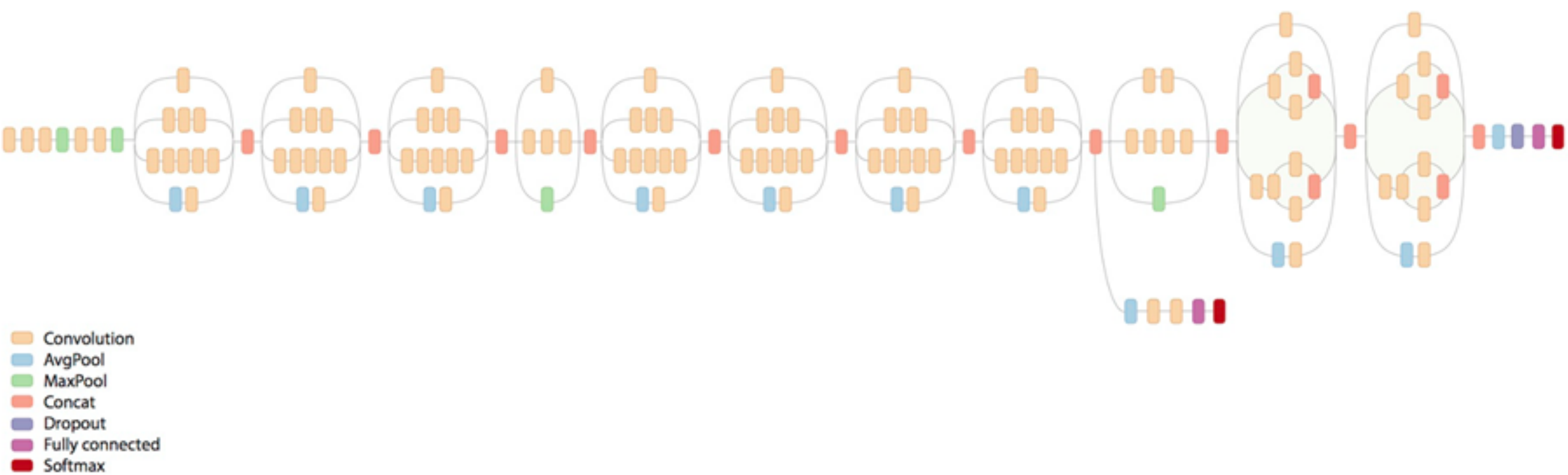
Training data	100s to 1000s of labeled images (small)
Computation	Moderate computation
Training Time	Seconds to minutes
Model accuracy	Good, depends on the pre-trained CNN model

Known Topology - ImageNet



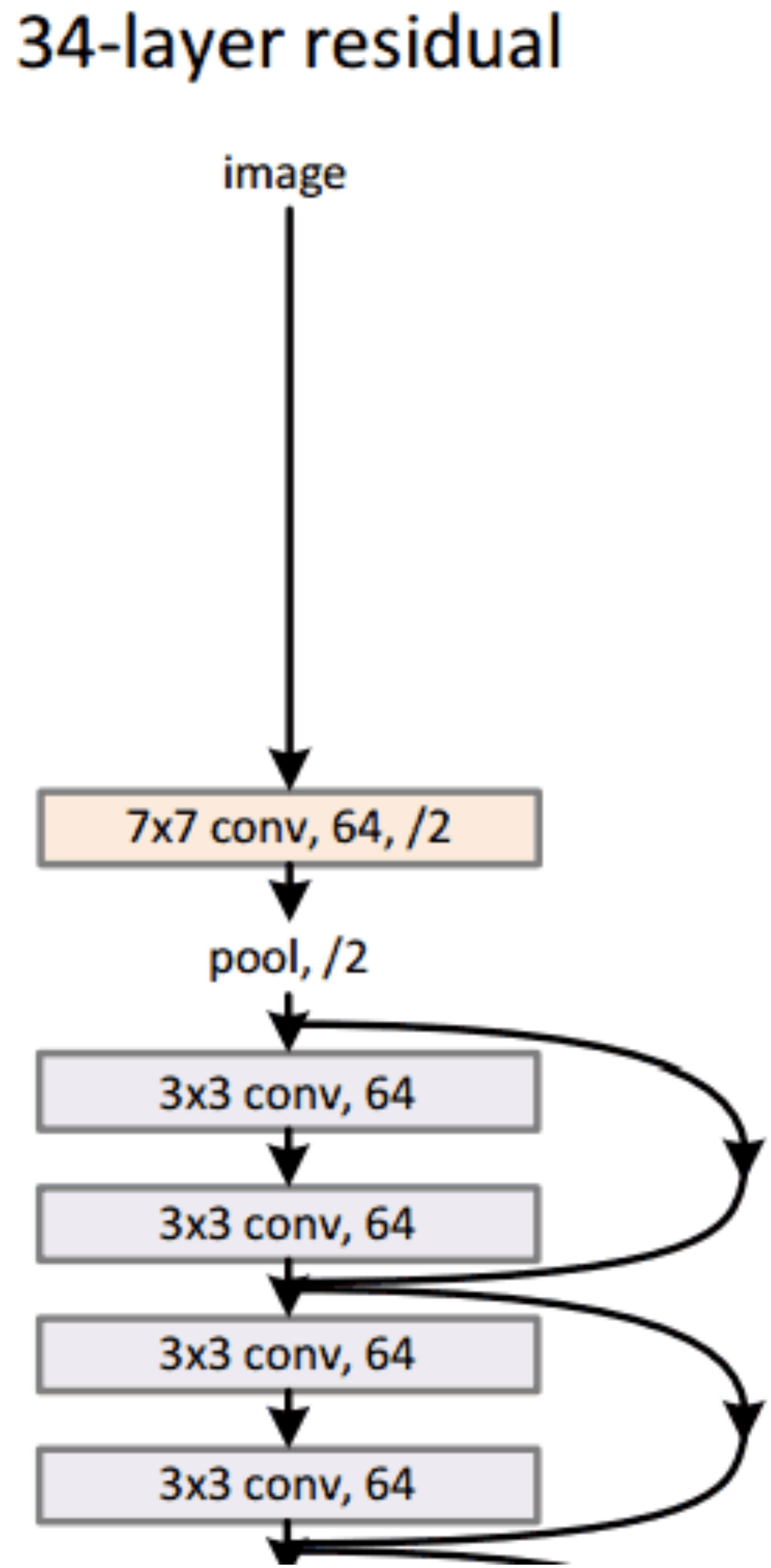
GoogLeNet



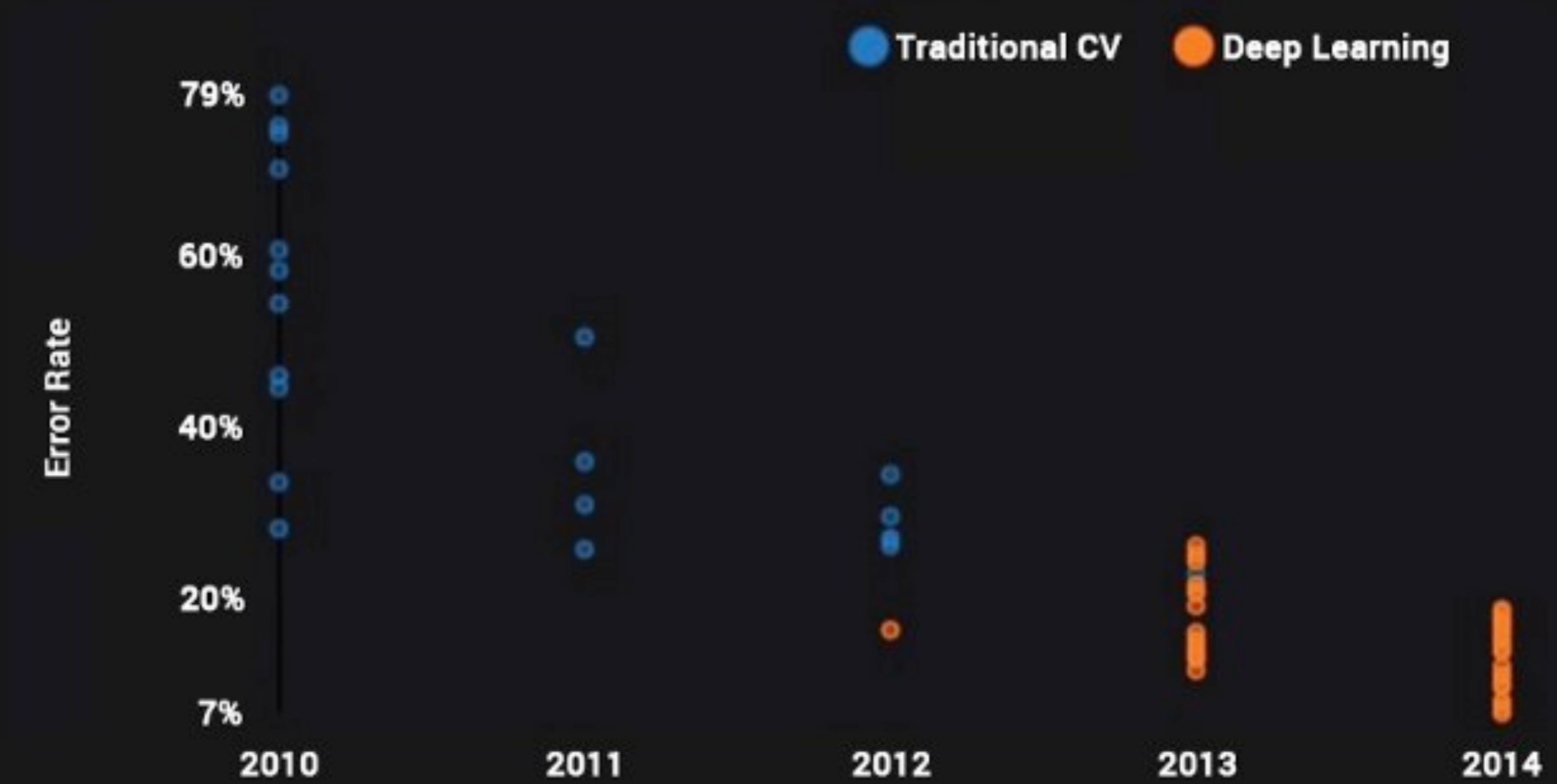


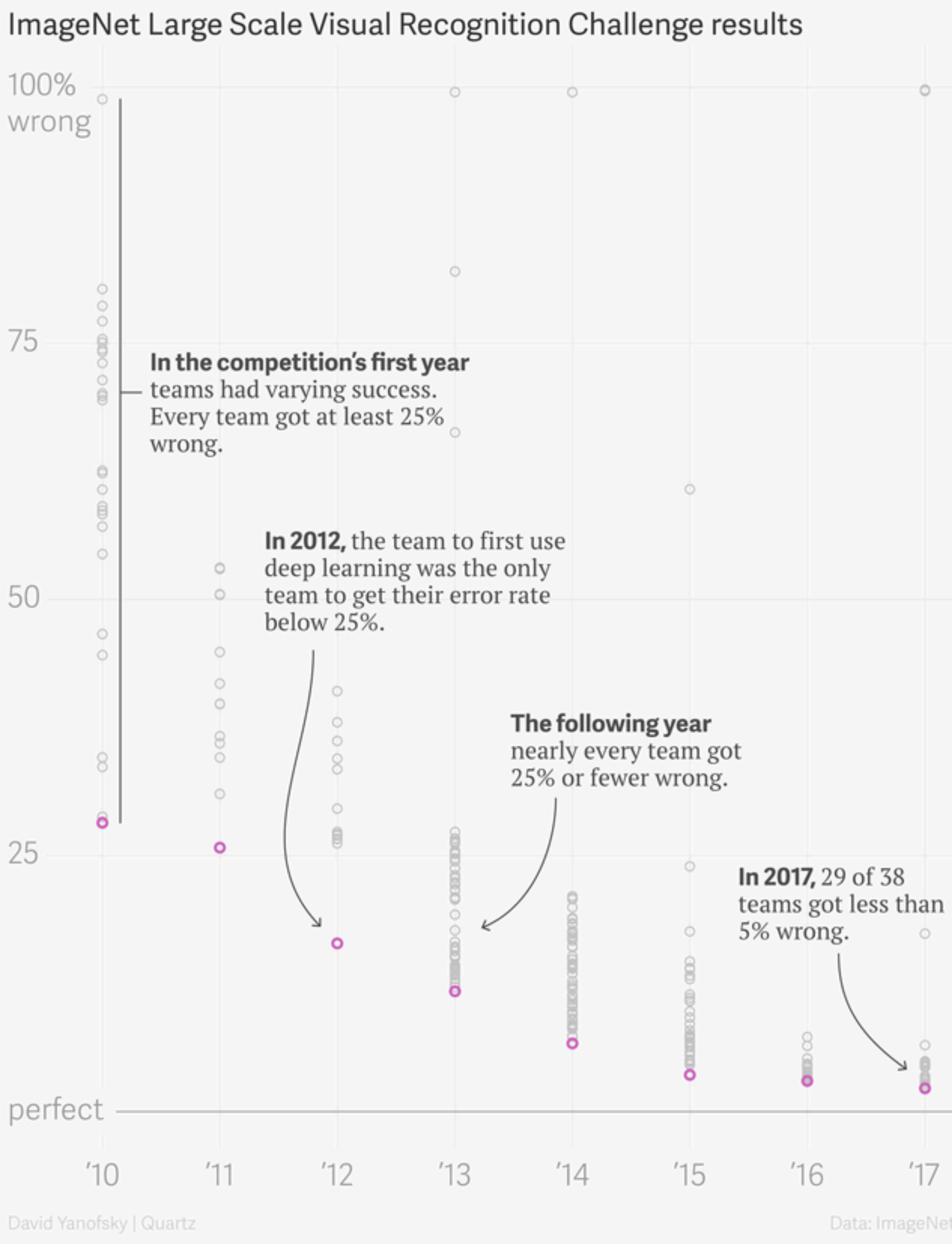
Another view of GoogLeNet's architecture.

ResNet


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64

ImageNet Error Rate 2010-2014





2010: 25+%

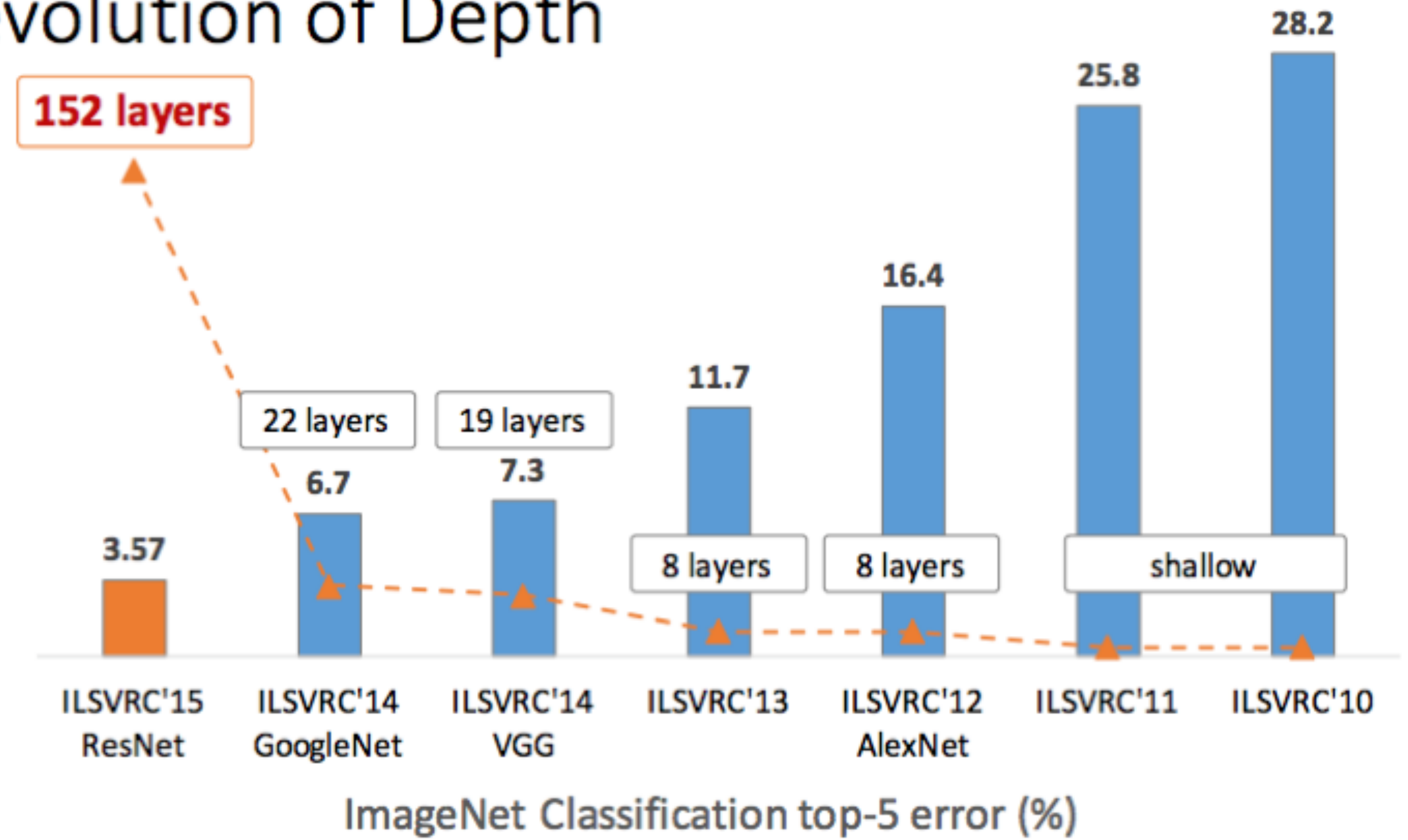
2012: 16% (AlexNet)

2015: 3.57% (ResNet)

2017: (29 out of 38 less than 5%)

https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-wo

Revolution of Depth

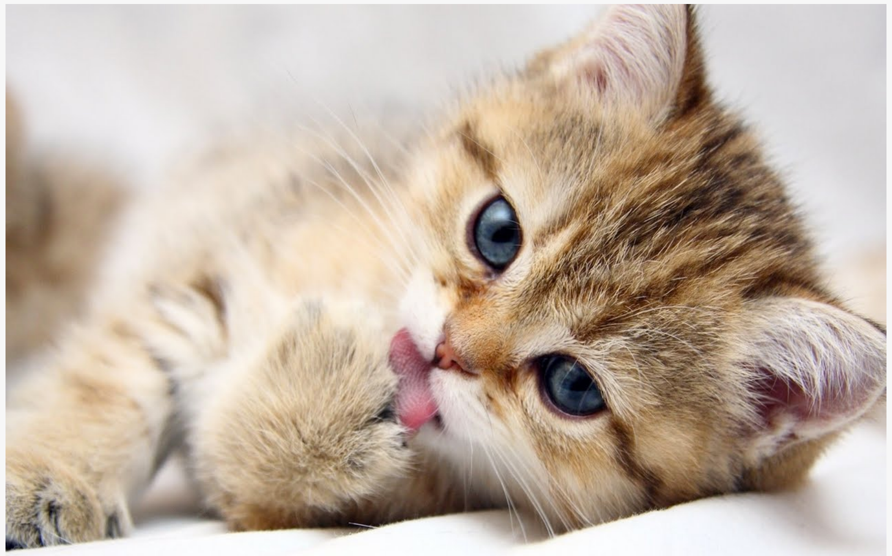


```python
from keras.applications.resnet50 import ResNet50
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input, decode_predictions
import numpy as np

model = ResNet50(weights='imagenet')

img = image.load_img('cat.jpg', target_size=(224, 224))
x = np.expand_dims(image.img_to_array(img), axis=0)
x = preprocess_input(x)

preds = model.predict(x)
print('Predicted:', decode_predictions(preds, top=3)[0])
```



```
('Predicted:', [(u'n02123045', u'tabby', 0.47442555), (u'n02127052', u'lynx', 0.39875993),
 (u'n02124075', u'Egyptian_cat', 0.048101209)])
```

Tabby 47.4%

Lynx 39.9%

Egyptian cat 4.8%

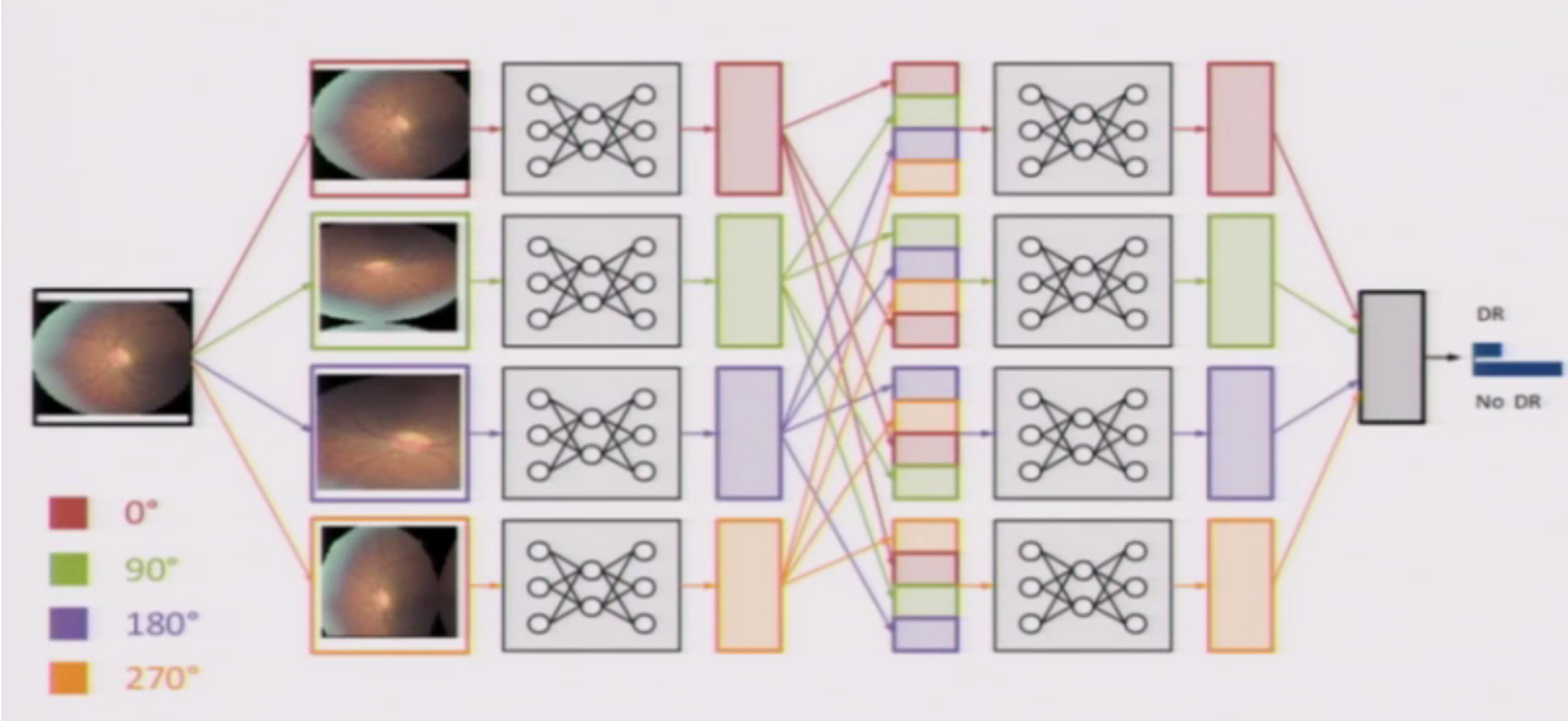


IMAGE CLASSIFICATION	MEAN ACCURACY
Our fine-tuned DCNN	**0.96**
Feature Based SVM	0.82

https://www.youtube.com/watch?v=nB15sW-mkHA

In Summary

if you can remember only three…

- CNN boost computer vision (but also NLP, language translation…)
- Deep learning means a lot of data with labels
- Reuse knowledge from model is key in future


DATA
WORKSHOP
20 PAŹDZIERNIKA O 19:00 BEZPŁATNY WEBINAR
Praktyczne uczenie maszynowe dla programistów
Wprowadzenie techniczne do narzędzi uczenia maszynowego
Dołącz na dataworkshop.eu/free-webinar

Praktyczne uczenie maszynowe dla programistów*

* Dla osób, które znają język Python (*jeden z najłatwiejszych języków dla początkujących)*

Uczenie poprzez **przykłady**, bez skomplikowanych i zawiłych (matematycznych) detali

Naucz się, jak zaimplementować **rozwiązanie end-to-end**

Start kursu: 30 października 2017 roku

12	14	31	44
Dni	Godzin	Minut	Sekund


DATA WORKSHOP
Praktyczne uczenie maszynowe dla programistów
-10%
Kod: lublin10
Na pakiety:
PREMIUM
GOLD

biznesmysli.pl

Vladimir Alekseichenko

Architect
General Electric

dataworkshop.eu

DATA
WORKSHOP

vova.me
slon1024
hello@vova.me

Thank you

@slon1024
hello@vova.me
biznesmysli.pl
dataworkshop.eu