W4995 Applied Machine Learning

Recap & Summary

04/29/19

Andreas C. Müller

Model inspection

Types of explanations

Explain "kind of model"

- How would the model change if this feature was dropped?
- Doesn't explain this particular fitted model

Explain model globally

- How does the output depend on the input?
- Often: some form of marginals

Explain model locally

- Why did it classify this point this way?

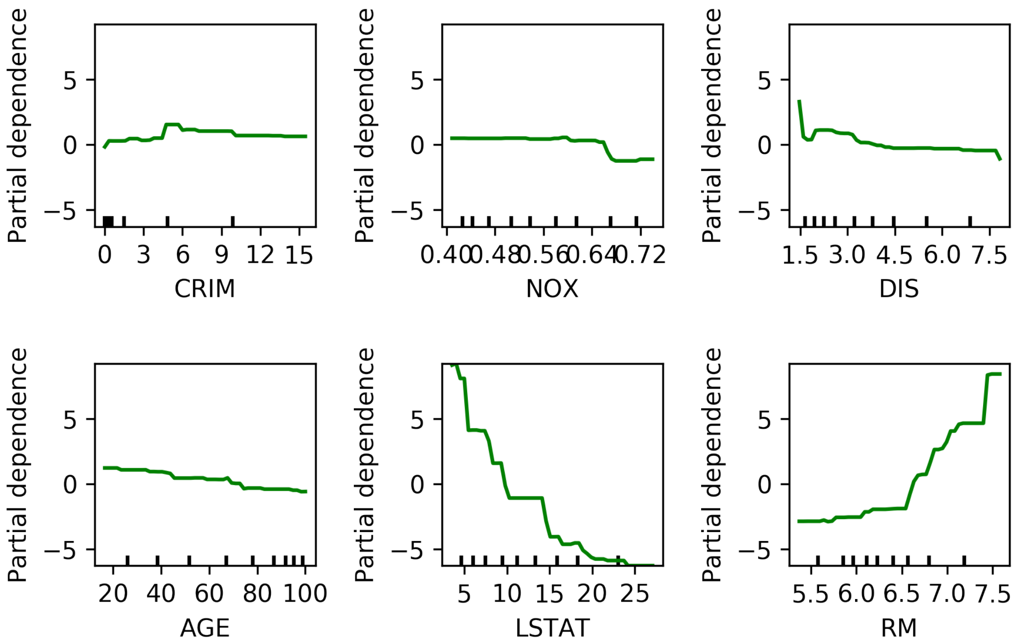
- Explanation could look like a "global" one but be different for each point.

Permutation importance

Idea: measure marginal influence of one feature

```
def permutation_importance(est, X, y, n_bootstrap=100):
  baseline_score = estimator.score(X, y)
  for f_idx in range(X.shape[1]):
      for b_idx in range(n_bootstrap):
          X_new = X.copy()
          X_new[:, f_idx] = np.random.shuffle(X[:, f_idx])
          feature_score = estimator.score(X_new, y)
          scores[f_idx, b_idx] = baseline_score - feature_score
```

Partial Dependence Plots



Feature selection

Why Select Features?

- Avoid overfitting (?)
- Faster prediction and training
- Less storage for model and dataset
- More interpretable model

AutoML

Formulating model-selection as Hyperparameter Optimization

- One big search, many conditional Hyper-Parameters
- Categorical, integer, continuous, conditional

$$\Lambda^* = \arg\max_{\Lambda} f(\Lambda)$$

Parameters Λ, model-evaluation f.

Approaches

Random Search

Bayesian Optimization, SMBO

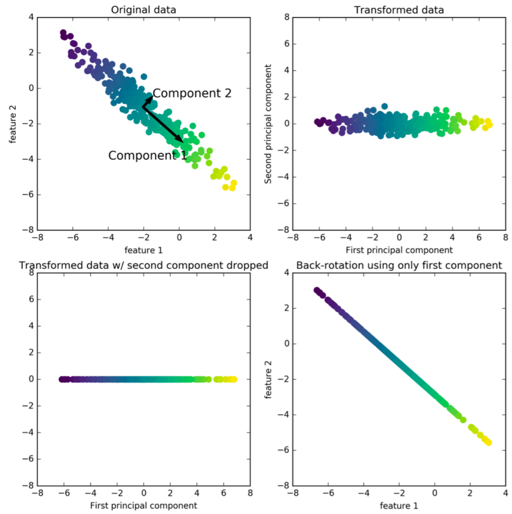
Successive Halving

Dimensionality Reduction

PCA

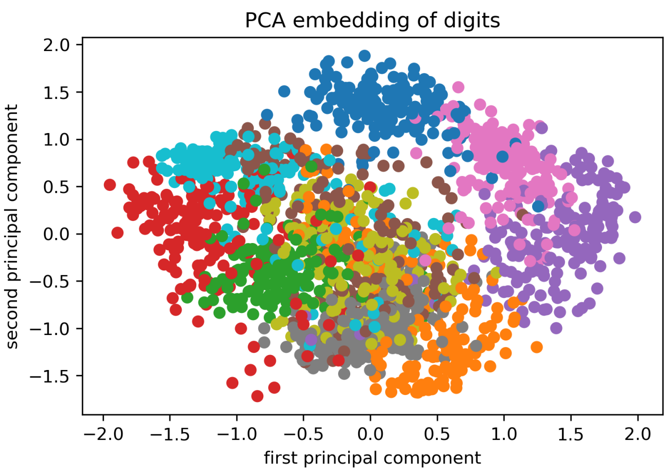
$$\max_{u_1 \in R^p, \|u_1\|=1} \operatorname{var}(X u_1)$$

$$\max_{u_1 \in R^p, \|u_1\|=1} u_1^T \operatorname{cov}(X) u_1$$

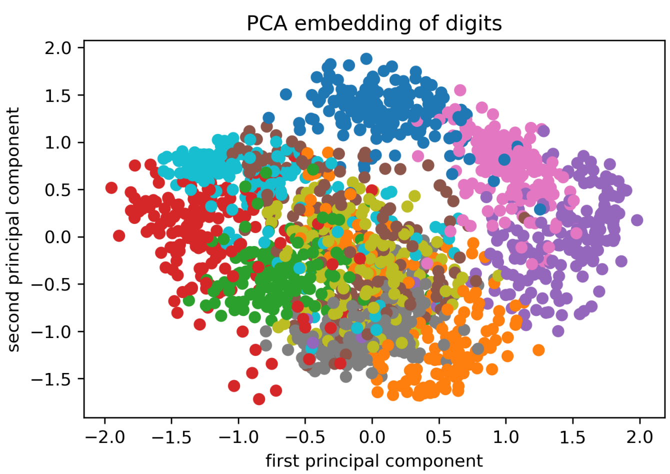


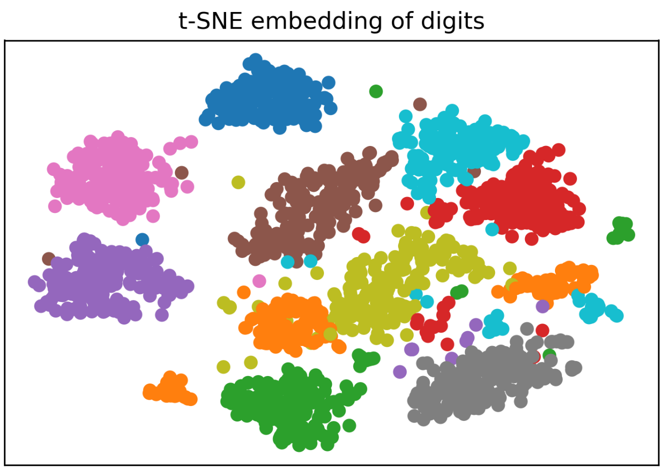
Manifold Learning

```python
from sklearn.manifold import TSNE
from sklearn.datasets import load_digits
digits = load_digits()
X = digits.data / 16.
X_tsne = TSNE().fit_transform(X)
X_pca = PCA(n_components=2).fit_transform(X)
```



```python
from sklearn.manifold import TSNE
from sklearn.datasets import load_digits
digits = load_digits()
X = digits.data / 16.
X_tsne = TSNE().fit_transform(X)
X_pca = PCA(n_components=2).fit_transform(X)
```





Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?

Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?
- Data Partitioning
  - Divide data by group before further processing

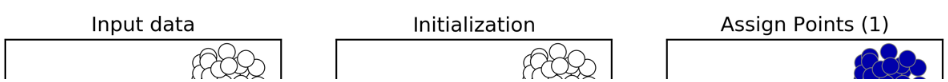
Clustering

- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?
- Data Partitioning
  - Divide data by group before further processing
- Unsupervised feature extraction
  - Derive features from clusters or cluster distances

Clustering

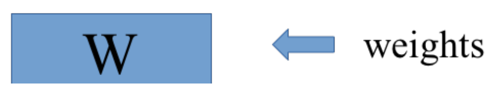
- Data Exploration
  - Are there coherent groups ?
  - How many groups are there ?
- Data Partitioning
  - Divide data by group before further processing
- Unsupervised feature extraction
  - Derive features from clusters or cluster distances
- Evaluation and parameter tuning
  - Quantitative measures of limited use
  - Usually qualitative measures used
  - Best: downstream tasks

K-Means algorithm



- Pick number of clusters k.

NMF



Text Data: Bag of Words

`“This is how you get ants.”`

N-grams: Beyond single words

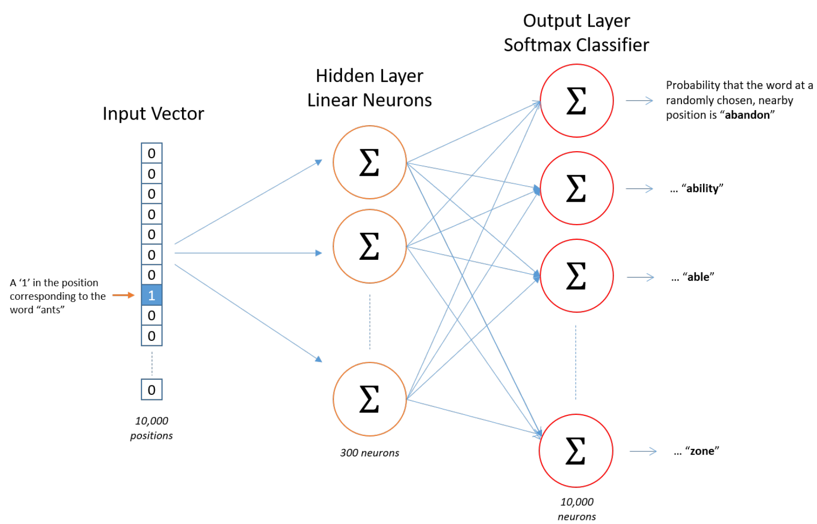
- Bag of words completely removes word order.
- "didn't love" and "love" are very different!

N-grams: Beyond single words

- Bag of words completely removes word order.
- "didn't love" and "love" are very different!

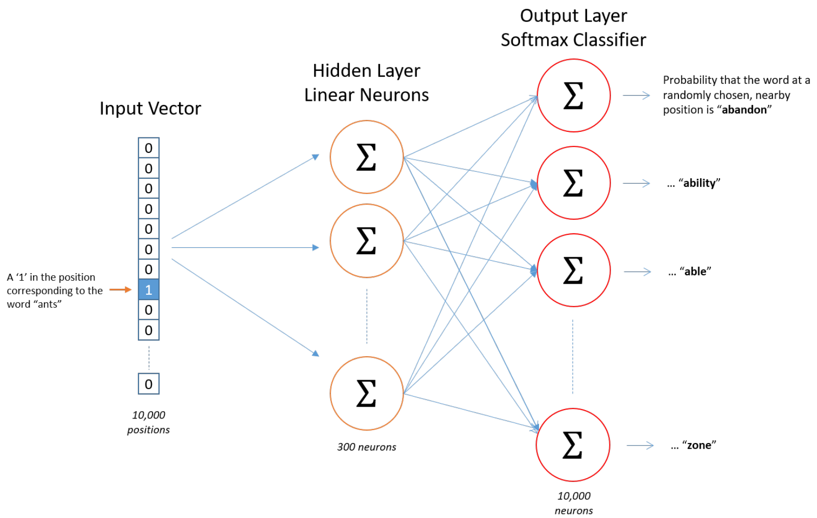
“This is how you get ants.”

Word Embeddings



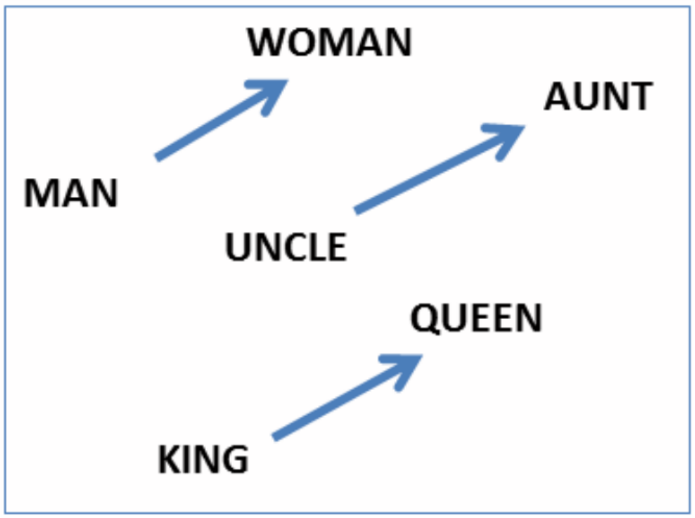
http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

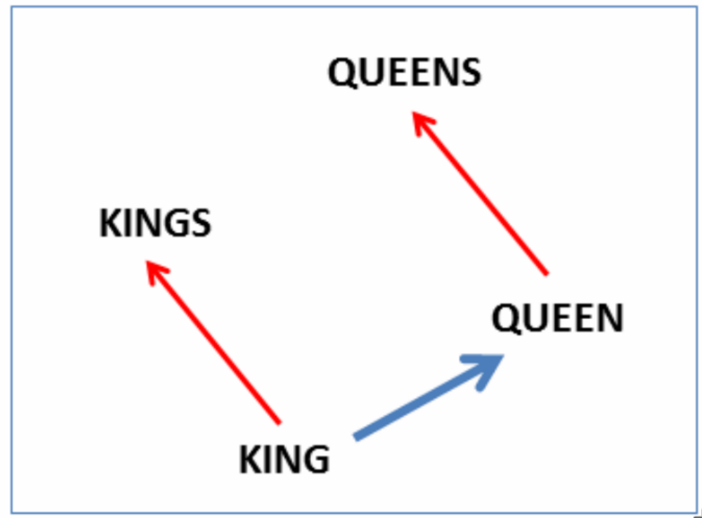
Word Embeddings



http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

Analogues and Relationships

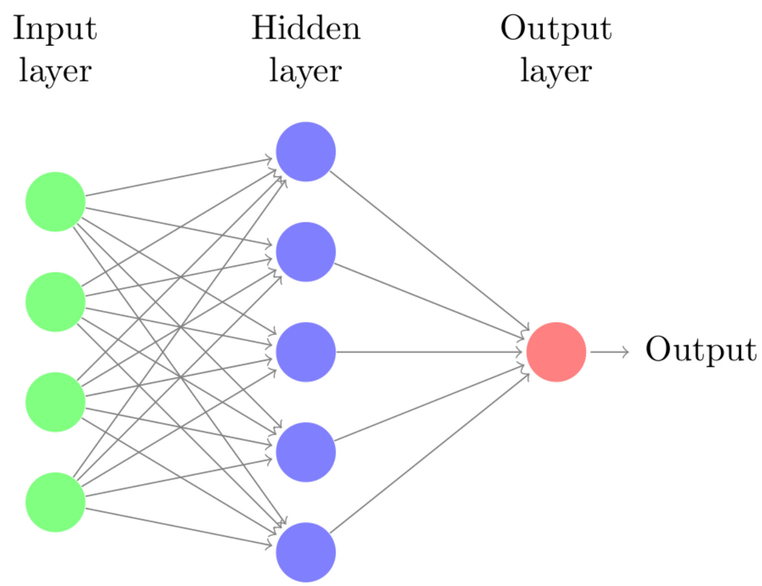




Answer “King is to Kings as Queen is to ?”:
Find closest vector to vec(“Queen”) + (vec(“Kings”) - vec(“King”))

[Mikolov et. al. Linguistic Regularities in Continuous Space Word Representations (2013)]

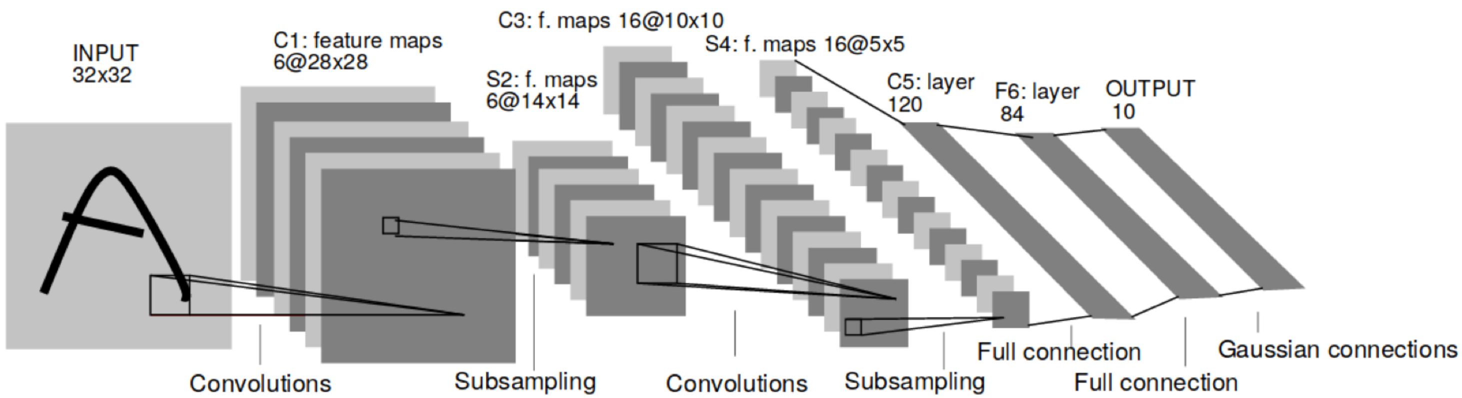
Neural Networks



$$h(x) = f(W_1 x + b_1)$$

$$o(x) = g(W_2 h(x) + b_2)$$

Convolution Neural Networks

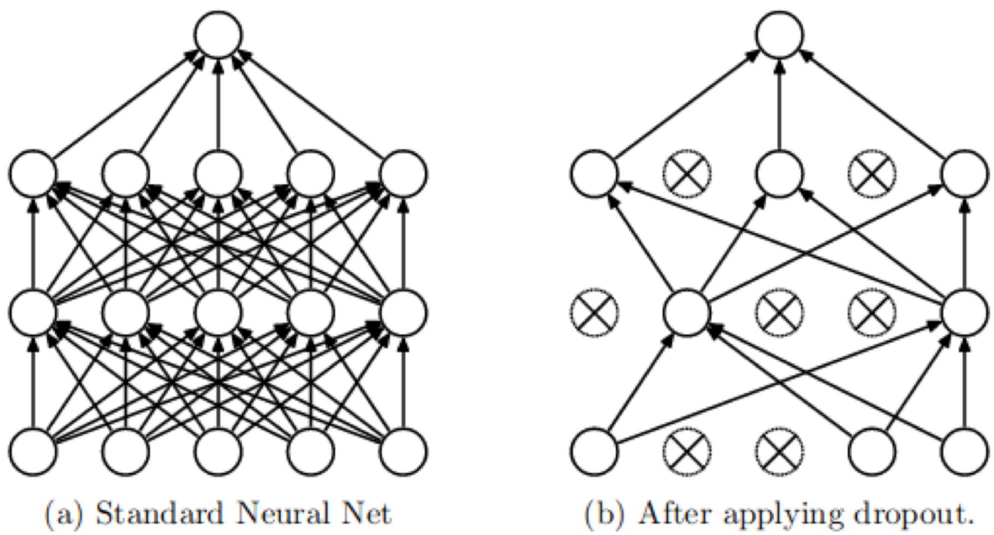


- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: Gradient-based learning applied to document recognition

28 / 38

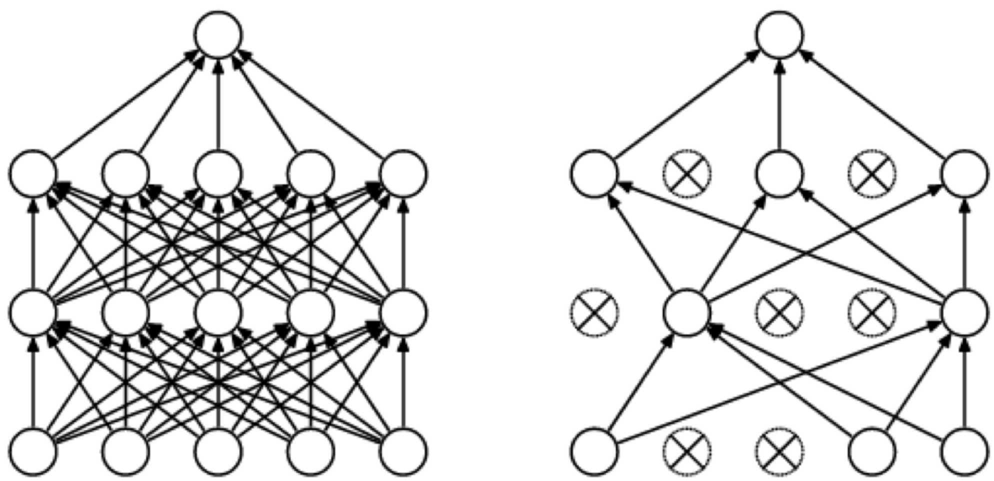
Tricks for learning Deep Nets

Drop-out Regularization



30 / 38

Drop-out Regularization



(a) Standard Neural Net (b) After applying dropout.

- https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

- Rate often as high as .5, i.e. 50% of units set to zero!
- Predictions: use all weights, down-weight by rate

31 / 38

Batch Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;

32 / 38

Problem

Solution: Residual Neural Networks

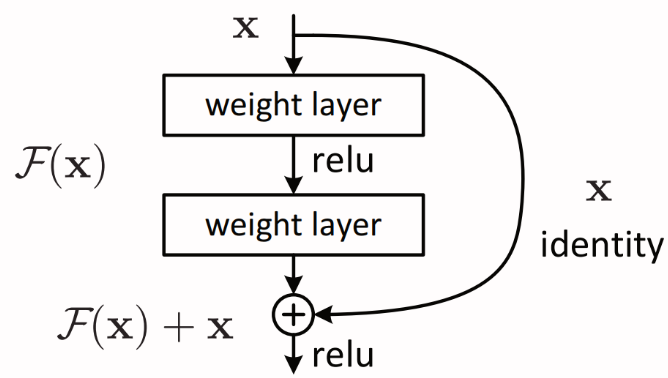


Figure 2. Residual learning: a building block.

$y = F(x, \{W_i\}) + x$ for same size layers

34 / 38

Solution: Residual Neural Networks

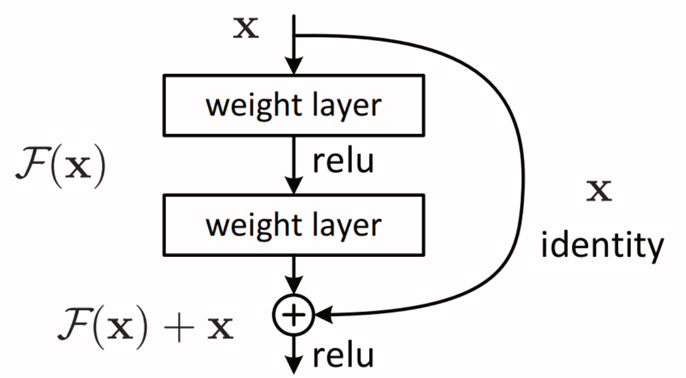
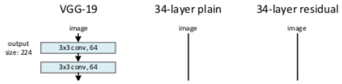


Figure 2. Residual learning: a building block.

$$y = F(x, \{W_i\}) + x \quad \text{for same size layers}$$

$$y = F(x, \{W_i\}) + W_s x \quad \text{for different size layers}$$


VGG-19
34-layer plain
34-layer residual
image
image
image
output
size: 224
3x3 conv, 64
3x3 conv, 64

Practical Considerations for Neural Nets

Questions ?