



# Attention in Deep Learning

**Alex Smola (smola@) and Aston Zhang (astonz@)**

**Amazon Web Services**

**ICML 2019, Long Beach, CA**

**[bit.ly/2R10hTu](https://bit.ly/2R10hTu)**

**[alex.smola.org/talks/ICML19-attention.key](https://alex.smola.org/talks/ICML19-attention.key)**

**[alex.smola.org/talks/ICML19-attention.pdf](https://alex.smola.org/talks/ICML19-attention.pdf)**



# Outline

## 1. Watson Nadaraya Estimator

## 2. Pooling

- Single objects - Pooling to attention pooling
- Hierarchical structures - Hierarchical attention networks

## 3. Iterative Pooling

Question answering / memory networks

## 4. Iterative Pooling and Generation

Neural machine translation

## 5. Multiple Attention Heads

- Transformers / BERT
- Lightweight, structured, sparse

## 6. Resources

# 1. Watson Nadaraya Estimator '64

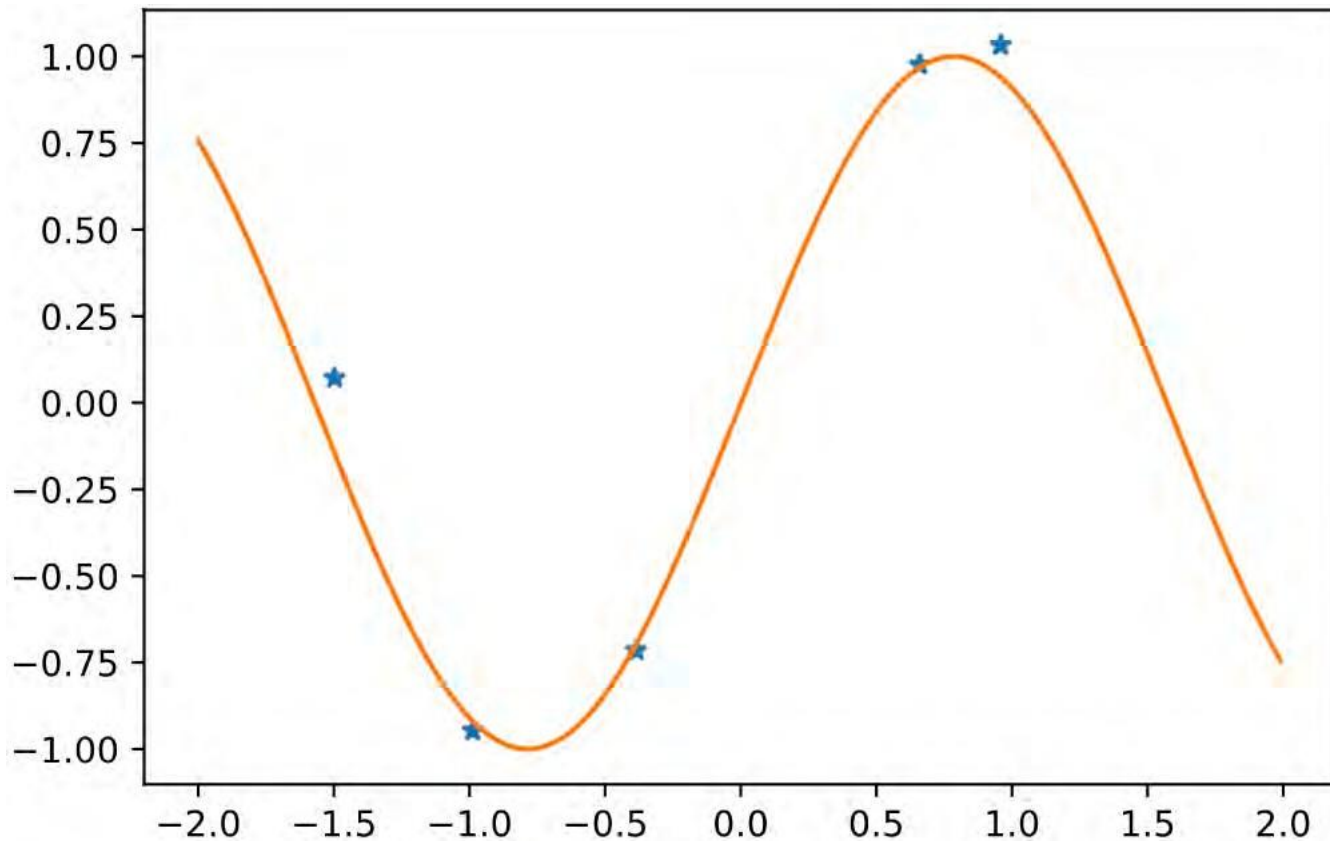


Geoffrey Watson



Elizbar Nadaraya

# Regression Problem



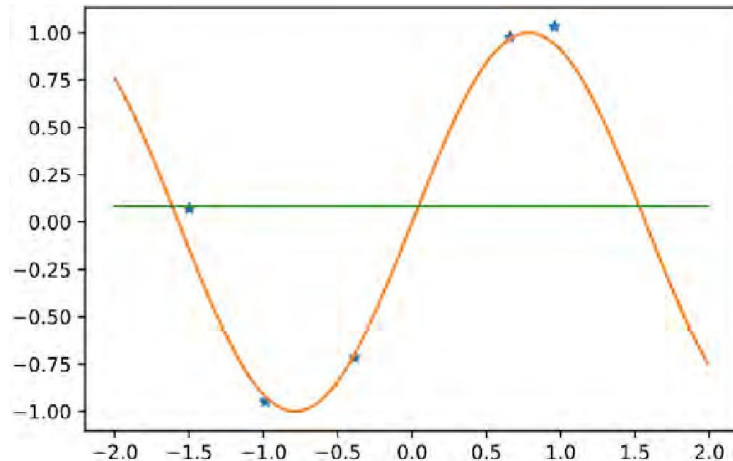
# Solving the regression problem

- Data  $\{x_1, \dots, x_m\}$  and labels  $\{y_1, \dots, y_m\}$
- Estimate label  $y$  at new location  $x$
- **The world's dumbest estimator**  
Average over all labels

$$y = \frac{1}{m} \sum_{i=1}^m y_i$$

- **Better idea (Watson, Nadaraya, 1964)**  
Weigh the labels according to location

$$y = \sum_{i=1}^m \alpha(x, x_i) y_i$$



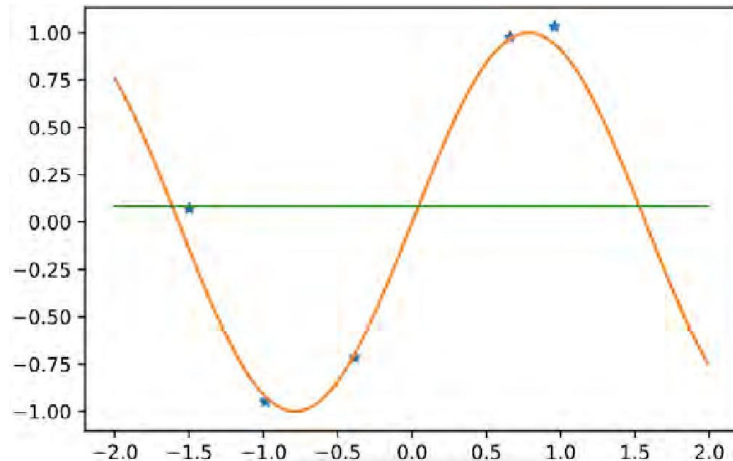
# Solving the regression problem

- Data  $\{x_1, \dots, x_m\}$  and labels  $\{y_1, \dots, y_m\}$
- Estimate label  $y$  at new location  $x$
- **The world's dumbest estimator**  
Average over all labels

$$y = \frac{1}{m} \sum_{i=1}^m y_i$$

- **Better idea (Watson, Nadaraya, 1964)**  
Weigh the labels according to location

$$y = \sum_{i=1}^m \alpha(x, x_i) y_i$$



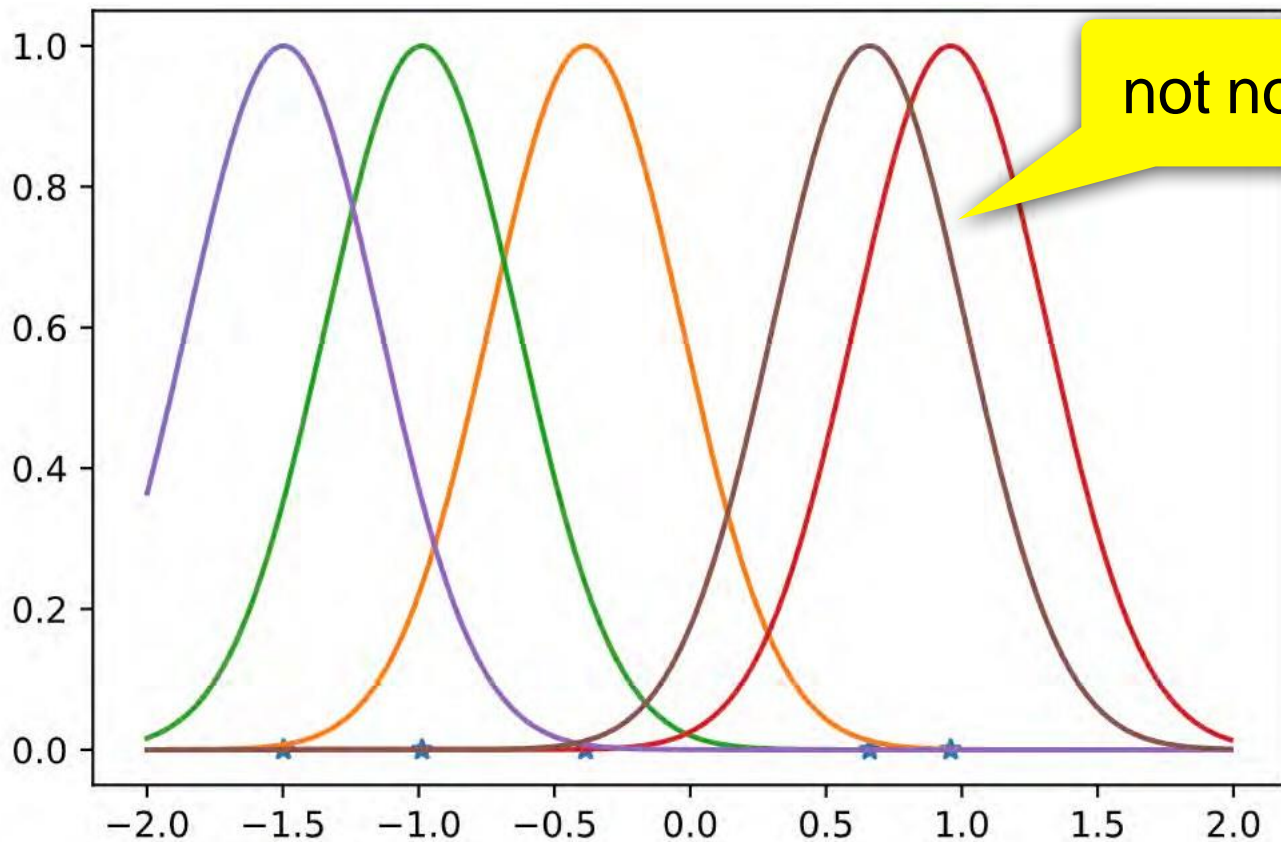
key

$\alpha(x, x_i) y_i$

query

value

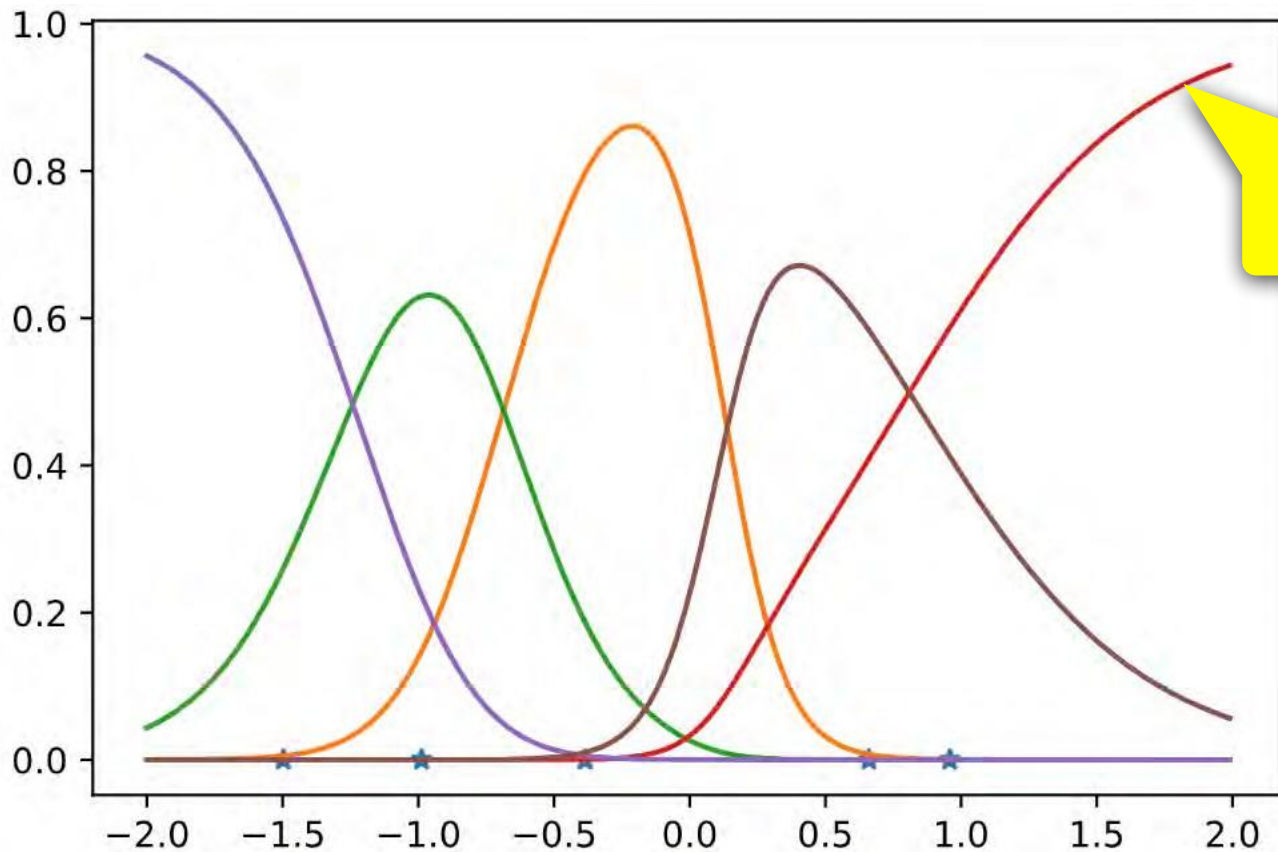
# Weighing the locations (e.g. with Gaussians)



not normalized

$$\alpha(x, x_i) \propto k(x_i, x)$$

# Weighing the locations (e.g. with Gaussians)

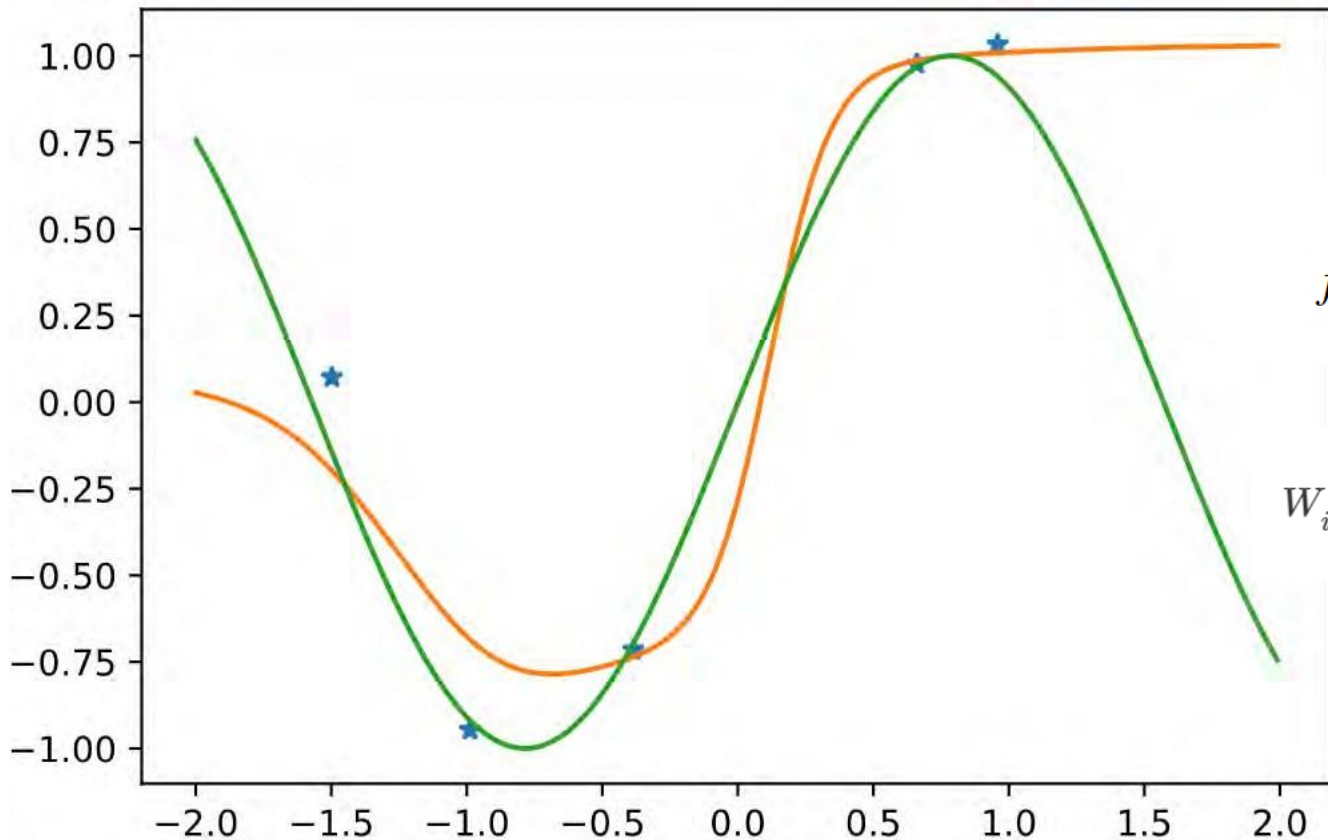


normalized

$$\alpha_i(x) = \frac{k(x_i, x)}{\sum_j k(x_j, x)}$$

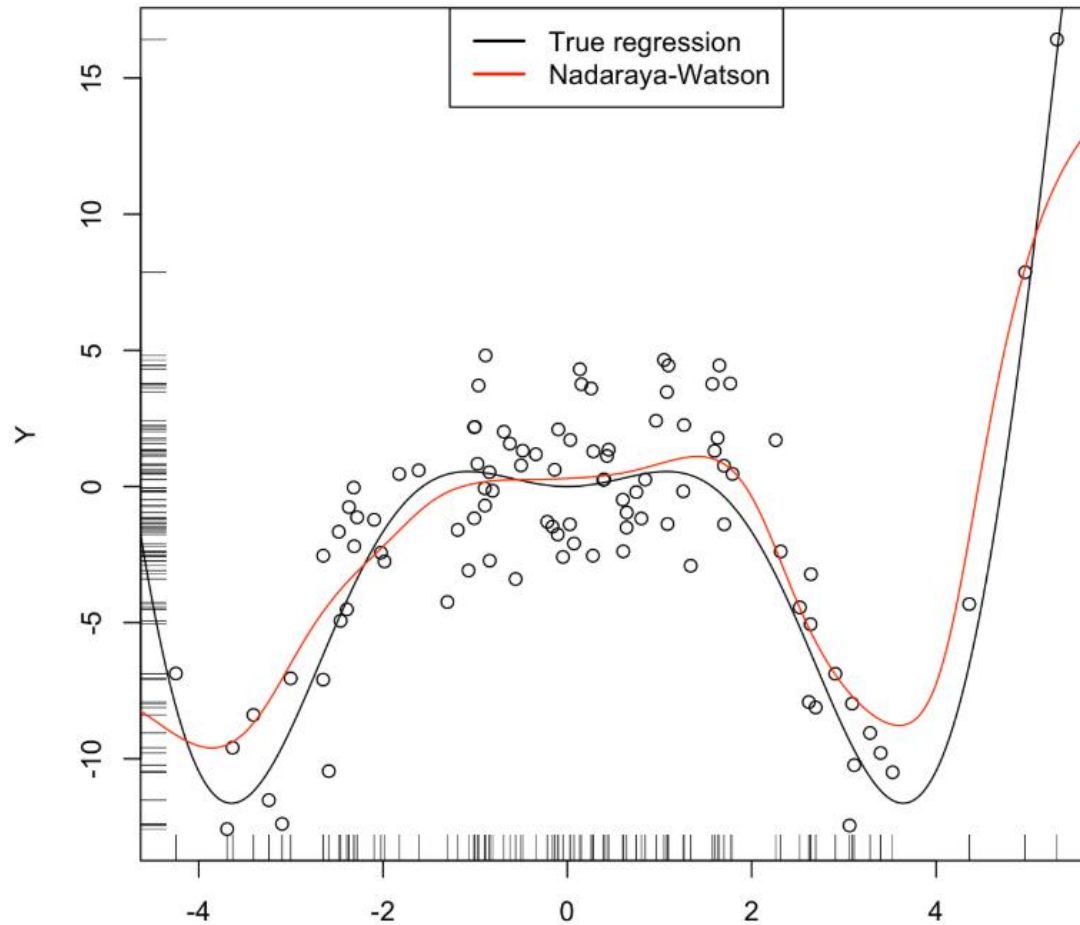


# Weighted regression estimate



$$f(x) = \sum_i y_i \frac{k(x_i, x)}{\sum_j k(x_j, x)}$$

$$W_i^0(x) := \frac{K_h(x - X_i)}{\sum_{i=1}^n K_h(x - X_i)}$$



# Why bother with a 55 year old algorithm?

- **Consistency**

Given enough data this algorithm converges to the optimal solution (can your deep net do this?)

- **Simplicity**

No free parameters - information is in the data not weights (or very few if we try to learn the weighting function)

# Why bother with a 55 year old algorithm?

- **Consistency**

Given enough data this algorithm converges to the optimal solution (can your deep net do this?)

- **Simplicity**

No free parameters - information is in the data not weights (or very few if we try to learn the weighting function)

- **Deep Learning Variant**

- Learn weighting function
- Replace averaging (pooling) by weighted pooling

A high-angle, close-up view of a swimming pool's surface. The water is a vibrant blue, and the pool is divided into lanes by dark blue lane lines. Each lane is further demarcated by a line of white and blue floating lane markers. The perspective is looking down the length of the pool, with the lane lines converging towards the top of the frame. The text '2. Pooling' is centered in the middle of the image in a white, sans-serif font.

## 2. Pooling

# Deep Sets (Zaheer et al. 2017)

- Deep (Networks on) Sets  $X=\{x_1, \dots, x_n\}$ 
  - Need **permutation invariance** for elements in set (e.g. LSTM doesn't work to ingest elements)
  - Theorem - all functions are of the form\*

$$f(X) = \rho \left( \sum_{x \in X} \phi(x) \right)$$

\*or some combination thereof

- Applications - point clouds, set extension, red shift for galaxies, text retrieval, tagging, etc.



# Deep Sets (Zaheer et al. 2017)

Outliers in sets - learn function  $f(X)$  on set such that

$$f(\{x\} \cup X) \geq f(\{x'\} \cup X) + \Delta(x, x')$$



# Deep Sets with Attention aka Multi-Instance Learning (Ilse, Tomczak, Welling, '18)

- Multiple Instance Problem  
Set contains one (or more) elements with desirable property (drug discovery, keychain). Identify those sets.
- Deep Sets have trouble focusing, hence weigh it

$$f(X) = \rho \left( \sum_{x \in X} \phi(x) \right) \longrightarrow f(X) = \rho \left( \sum_{x \in X} \alpha(w, x) \phi(x) \right)$$

- Attention function e.g.  $\alpha(w, x) \propto \exp(w^\top \tanh Vx)$



# Deep Sets

## ① Permutation Invariant

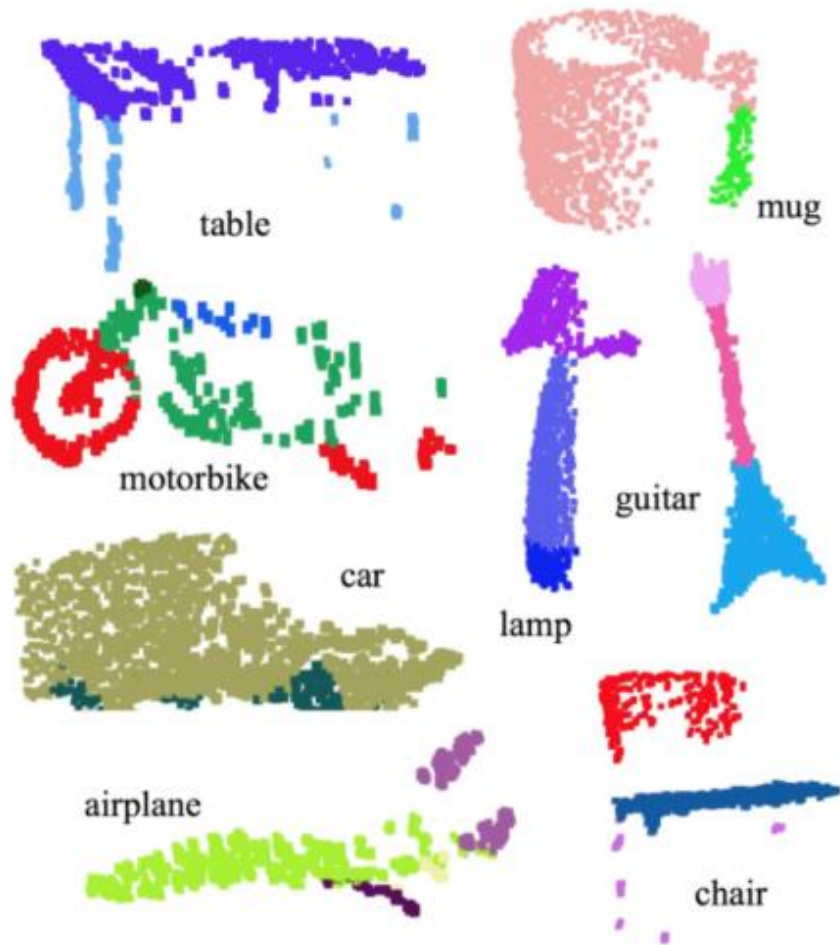
$$f([\text{man}, \text{woman}, \text{man}]) = [\text{apple}, \text{banana}]$$

$$f([\text{woman}, \text{man}, \text{man}]) = [\text{apple}, \text{banana}]$$

⋮

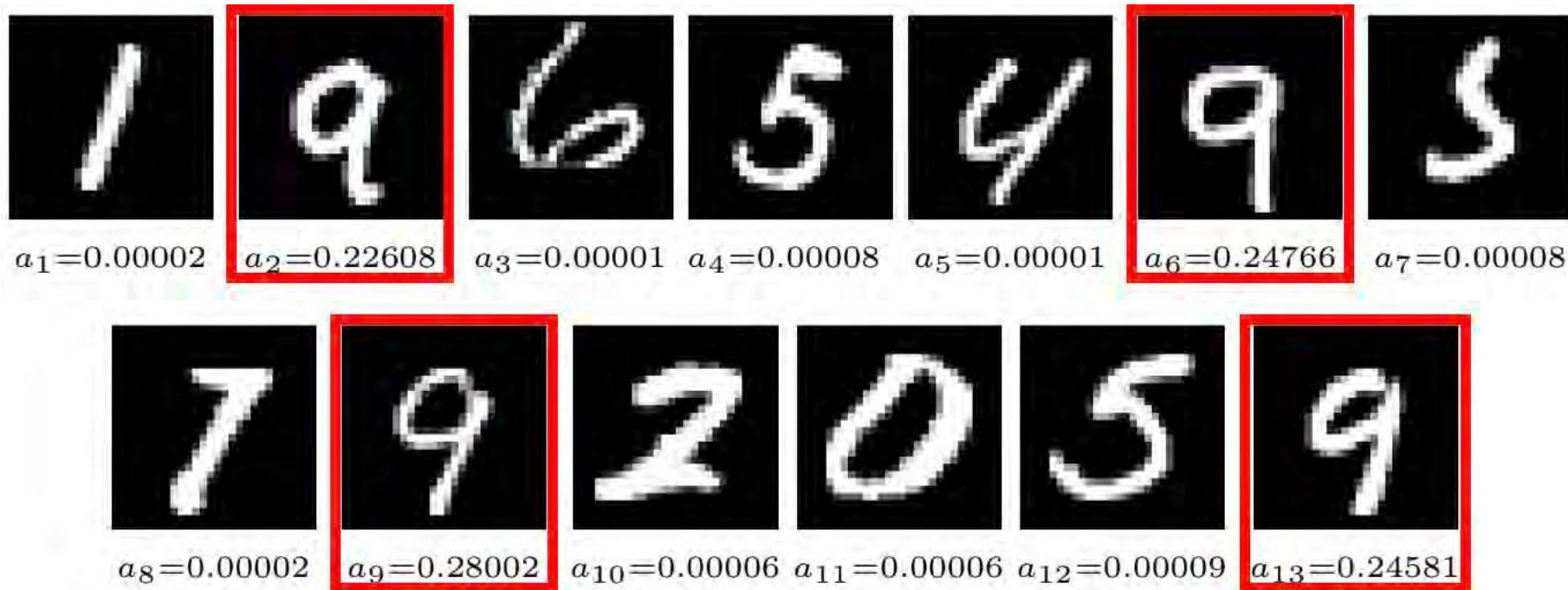
$$f([\text{man}, \text{man}, \text{woman}]) = [\text{apple}, \text{banana}]$$

# point clouds

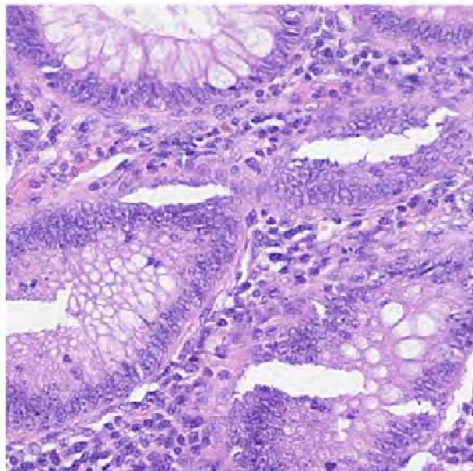


# Deep Sets with Attention aka Multi-Instance Learning (Ilse, Tomczak, Welling, '18)

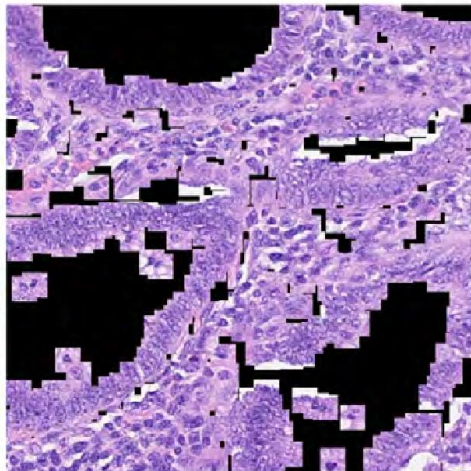
Identifying sets that contain the digit '9'



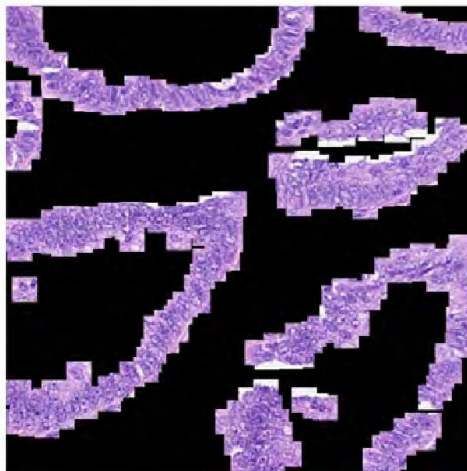
# Deep Sets with Attention aka Multi-Instance Learning (Ilse, Tomczak, Welling, '18)



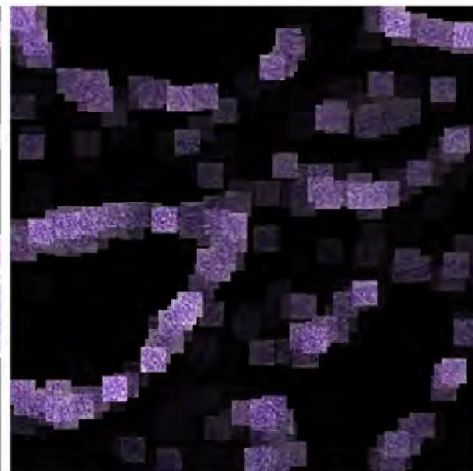
tissue  
sample



windowed  
cell nuclei



cancerous  
cells



attention  
weights

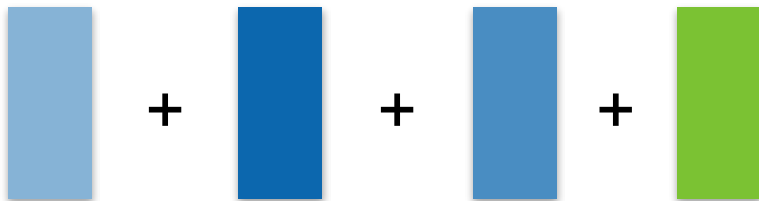
# Bag of words (Salton & McGill, 1986)

## Word2Vec (Mikolov et al., 2013)

- Embed each word in sentence (word2vec, binary, GRU ...)
- Add them all up
- Classify

$$f(X) = \rho \left( \sum_{i=1}^n \phi(x_i) \right)$$

The tutorial is awesome.



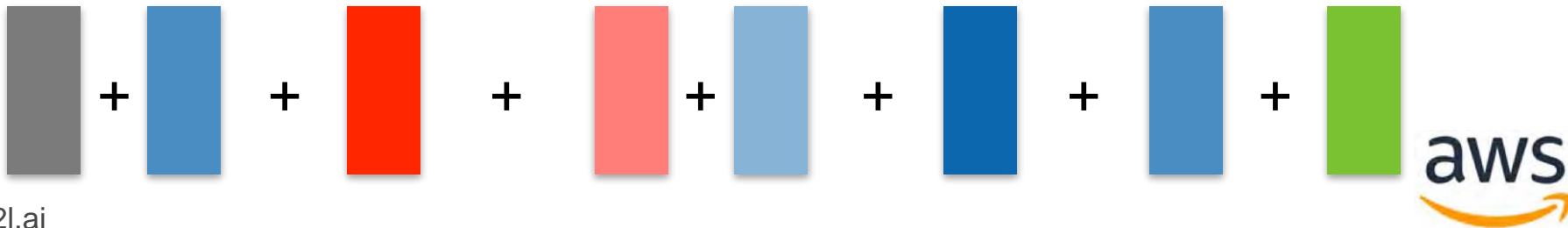
# Bag of words (Salton & McGill, 1986)

## Word2Vec (Mikolov et al., 2013)

- Embed each word in sentence (word2vec, binary, GRU ...)
- Add them all up
- Classify

$$f(X) = \rho \left( \sum_{i=1}^n \phi(w_i) \right)$$

Alex is obnoxious but the tutorial is awesome.



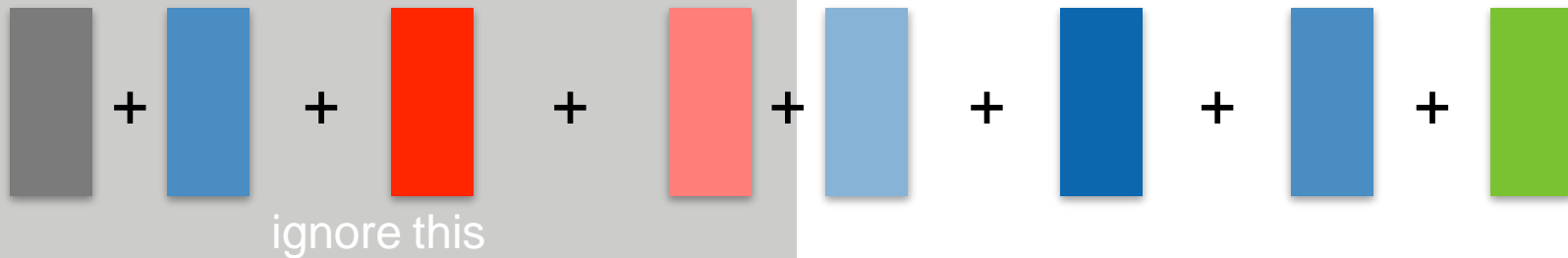
# Bag of words (Salton & McGill, 1986)

## Word2Vec (Mikolov et al., 2013)

- Embed each word in sentence (word2vec, binary, GRU ...)
- Add them all up
- Classify

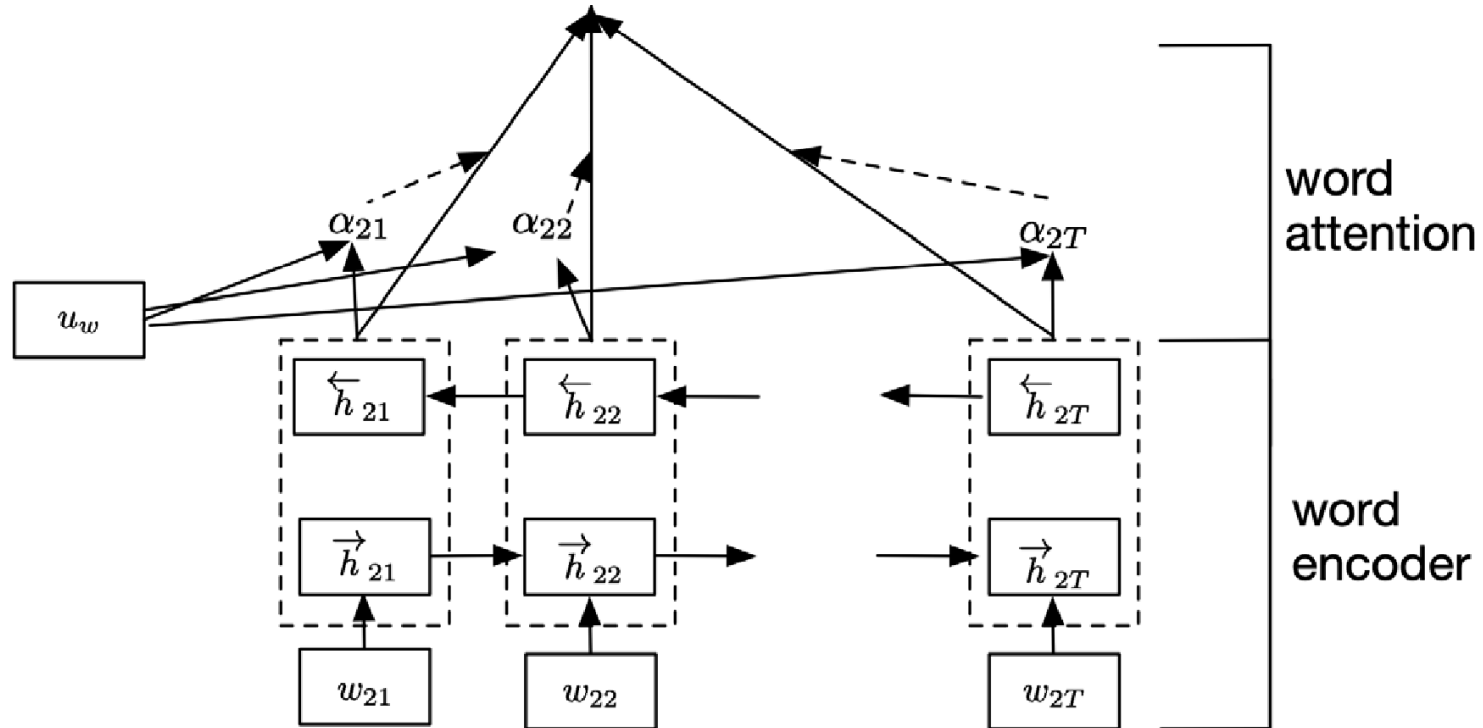
$$f(X) = \rho \left( \sum_{i=1}^n \phi(w_i) \right)$$

Alex is obnoxious but the tutorial is awesome.



# Attention weighting for documents (Wang et al, '16)

$$f(X) = \rho \left( \sum_{i=1}^n \phi(w_i) \right) \longrightarrow f(X) = \rho \left( \sum_{i=1}^n \alpha(w_i, X) \phi(w_i) \right)$$





# Hierarchical attention weighting (Yang et al. '17)

Some sentences are more important than others ...

GT: 4 Prediction: 4

pork belly = delicious .  
scallops ?  
i do n't .  
even .  
like .  
scallops , and these were a-m-a-z-i-n-g .  
fun and tasty cocktails .  
next time i 'm in phoenix , i will go  
back here .  
highly recommend .

GT: 0 Prediction: 0

terrible value .  
ordered pasta entree .  
.  
\$ 16.95 good taste but size was an  
appetizer size .  
.  
no salad , no bread no vegetable .  
this was .  
our and tasty cocktails .  
our second visit .  
i will not go back .

# Hierarchical attention

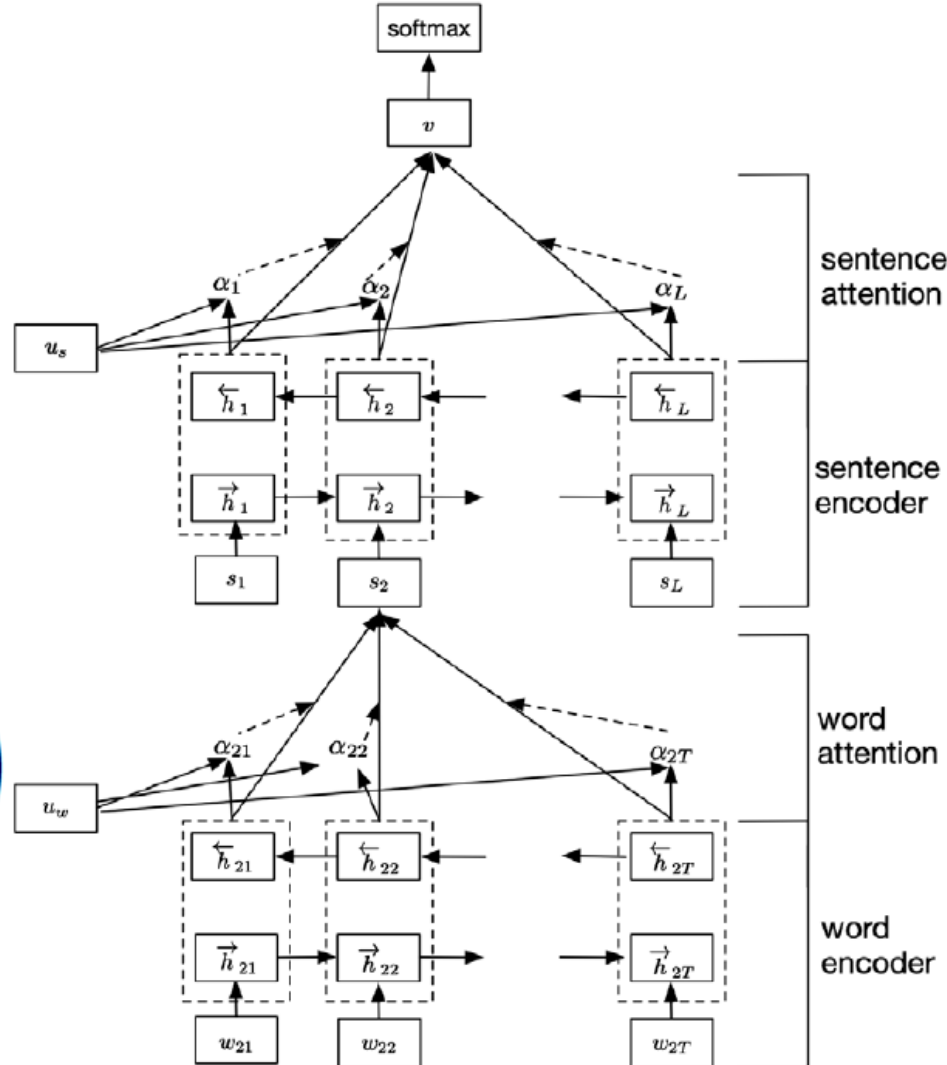
- Word level

$$f(s_i) = \rho \left( \sum_{j=1}^{n_i} \alpha(w_{ij}, s_i) \phi(w_{ij}) \right)$$

- Sentence level

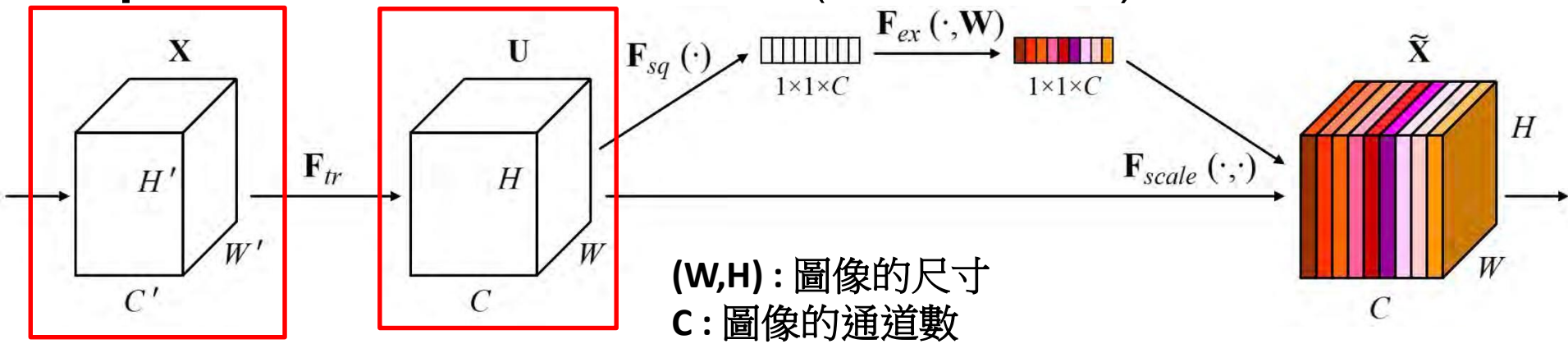
$$g(d) = \rho \left( \sum_{i=1}^n \alpha(s_i, d) \phi(f(s_i)) \right)$$

- Embeddings e.g. via GRU



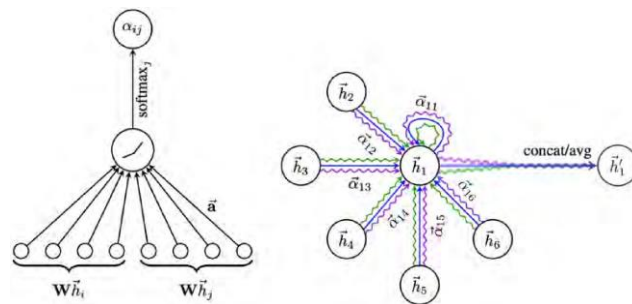
# More Applications

## Squeeze Excitation Networks (Hu et al., '18)



Feature map

## Graph Attention Networks (Velickovic et al., '18)



# Attention Summary

- Pooling

$$f(X) = \rho \left( \sum_{x \in X} \phi(x) \right)$$

Query  $w$  can depend on context

- Attention pooling

$$f(X) = \rho \left( \sum_{x \in X} \alpha(x, w) \phi(x) \right)$$

- Attention function (normalized to unit weight) such as

$$\alpha(x, X) \propto \exp \left( w^T \tanh Ux \right)$$