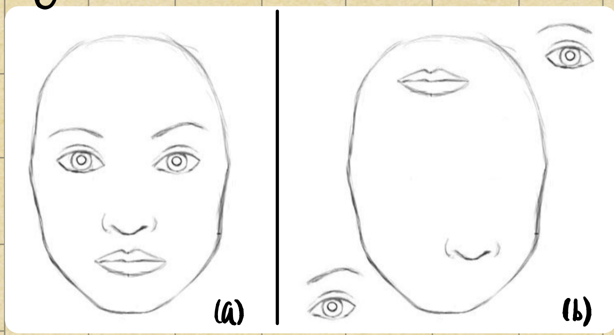
# 0. Background Introduction




① Internal data representation of a convolutional neural network does not take into account important spatial hierarchies between simple and complex objects.

② CNN need too much data.

③ part of image? ★★★★★

Hinton's revealations: Inverse graphics

人类的视网膜只接收到二维讯息，但却可以从中解构出层次表示 hierarchical representation. 从而想像出某物体的三维图像. 那么如何让机器也能学到层次表示呢? (可识别多角度的同一物体)

Invariance: by changing the input a little, the output still stays the same

* vectors encapsulate all important information about the state of the features.

# 1. Capsule 是什么?

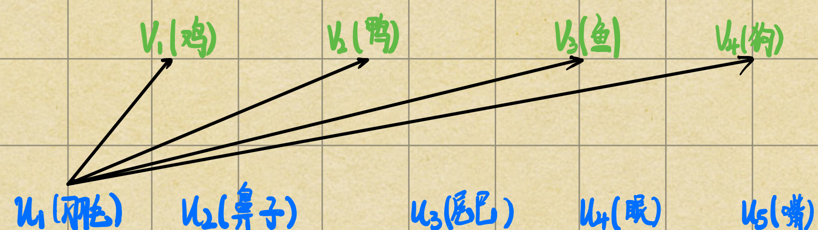
Core idea: "vector in vector out" rather than "scalar in scalar out" & add cluster into network.

Neuron ⟶ scalar

Capsule ⟶ vector

& output is a result of input's cluster.

Example: classification




y, u

$$(p_{1|1}, p_{2|1}, p_{3|1}, p_{4|1}) = \frac{1}{Z_1}\left(e^{\langle u_1, v_1\rangle}, e^{\langle u_1, v_2\rangle}, e^{\langle u_1, v_3\rangle}, e^{\langle u_1, v_4\rangle}\right)$$

∴对第i个特征($u_i$)有$(p_{1|i}, p_{2|i}, p_{3|i}, p_{4|i})$

why not:

$$(p_{1|1}u_1, p_{2|1}u_1, p_{3|1}u_1, p_{4|1}u_1) = \frac{u_1}{Z_1}\left(e^{\langle u_1, v_1\rangle}, e^{\langle u_1, v_2\rangle}, e^{\langle u_1, v_3\rangle}, e^{\langle u_1, v_4\rangle}\right) = V_1$$

$$\Rightarrow V_j = squash\left(\sum_i p_{j|i} \cdot u_i\right) = squash\left(\sum_i \frac{e^{\langle u_i, v_j\rangle}}{Z_i} \cdot u_i\right)$$

聚类中心　　　　特征分类

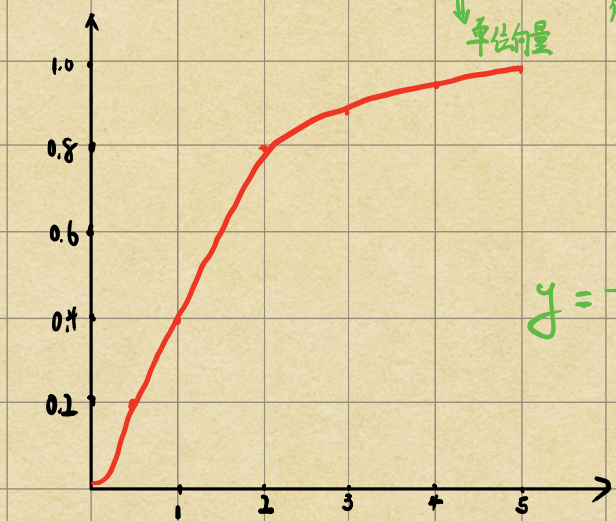
What is squash?

胶囊的模长代表这个特征的显著程度. 衡量"显著程度"使用 squash

$$Squash(x) = \frac{\|x\|^2}{1+\|x\|^2} \cdot \frac{x}{\|x\|}$$

单位向量

: 若图片中的特征(眼.鼻...)有了轻微的变化, 也仅仅是向量的方向变了, 而向量的模长(概率/显著程度)没有改变!




$$y = \frac{x^2}{1+x^2}$$

Dynamic Routing: 放弃梯度下降!

$$(p_{1|1}u_1, p_{2|1}u_1, p_{3|1}u_1, p_{4|1}u_1) = \frac{u_1}{Z_1}\left(e^{\langle u_1, v_1\rangle}, e^{\langle u_1, v_2\rangle}, e^{\langle u_1, v_3\rangle}, e^{\langle u_1, v_4\rangle}\right) = V_1$$

$V_i \longrightarrow$ softmax $\longrightarrow V_i$ How to deal with it?

Iteration method (Dynamic Routing)

Example: Given $(x_1, x_2, \cdots, x_n)$ to get a encoded $x$.

$$X = \sum_i \lambda_i X_i$$

$$X = \sum_i \frac{e^{\langle x, x_i\rangle}}{Z} \cdot x_i$$

$x = \bar{x}$
$\downarrow$
$x'$
$\downarrow$
$x''$ .....

Dynamic Routing Algorithm: ← Hardest part!!!

① intial $b_{ij} = 0$ for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l+1)$

② for r iterations do: where $b_{ij}$ means $\langle u_i, v_j\rangle$ i:特征 j:类别

$c_i \leftarrow \mathrm{softmax}(b_i)$ $c_i$ means $\sum_j e^{\langle u_i, v_j\rangle} / z_i$

$s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$ $\hat{u}_{j|i}$ means $u_i \cdot W_{ij}$

$v_j \leftarrow \mathrm{squash}(s_j)$

$b_{ij} \leftarrow \langle \hat{u}_{j|i}, v_j\rangle$

Summary:

1. 通过聚类来组合特征 ⟶ 人类使用自己的方式或熟悉的事物(底层特征)去理解新事物(特征组合)

2. Neural Network: $\mathrm{scalar}(h_j) = f(\sum_i w_i \cdot x_i + b)$

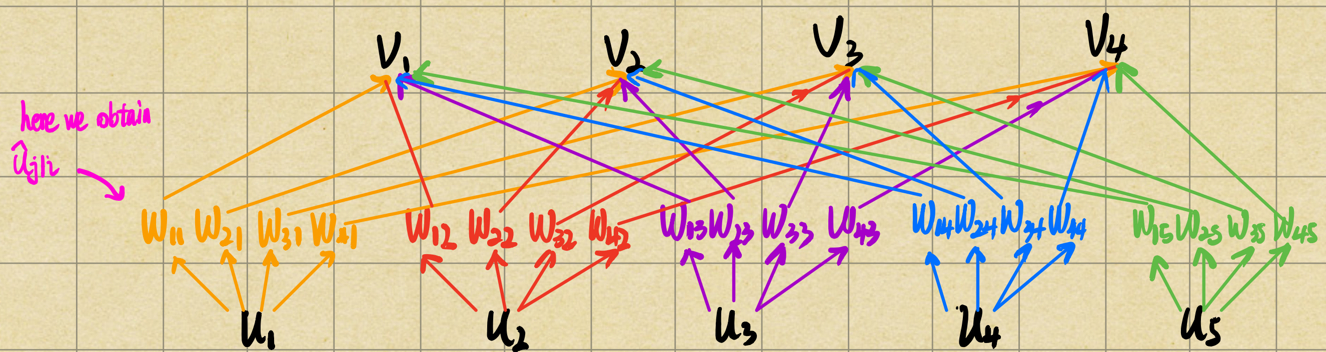
   Capsule: $\mathrm{vector}(v_j) = \mathrm{squash}\left(\sum_i \frac{e^{(W_{ij}\cdot u_i, v_j)}}{z_i} \cdot W_{ij} \cdot u_i\right)$ (without bias interesting)

3. $W_{ij}$ encode relationship (spatial etc.) between features

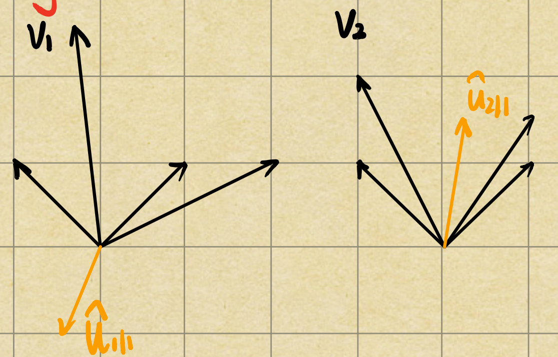
## 2. Keras 代码实现

### ①全连接动态路由(fully-connected dynamic routing)

$$v_j = \mathrm{squash}\left(\sum_i \frac{e^{(\hat{u}_{j|i}, v_j)}}{z_i} \cdot \hat{u}_{j|i}\right), \text{ where } \hat{u}_{j|i} = W_{ji} \cdot u_i.$$



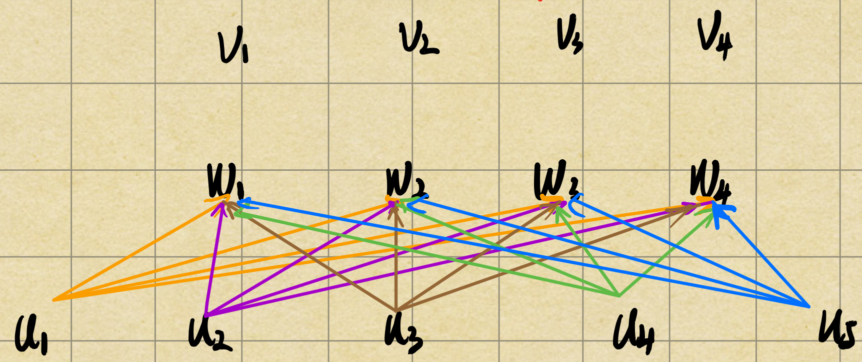

how $W_{ij}$ work?




$\therefore c_{11}\downarrow, c_{12}\uparrow$, then $W_{11}\cdot u_1 \downarrow$ $W_{12}\cdot u_1 \uparrow$

## ② 共享权值动态路由 $W_{ji}=W_j$

针对特征($u_i$)输入数量不确定的情形: (make sense: CNN权值共享)




$$V_j=\text{squash}\left(\sum_i \frac{e^{(\hat{u}_{j|i},V_j)}}{Z_i}\cdot \hat{u}_{j|i}\right),\quad \hat{u}_{j|i}=W_j\cdot u_i$$

## 3. Future work:

① Squash函数的改进

② 更加 make sense 的解释.

③ Capsule网络在其它领域上的应用

# 4. 原理再探：

① K-means 聚类： $u_1, u_2, \cdots, u_n \longrightarrow k$ classes

find $v_1, v_2, \cdots, v_k$ to $L=\sum_{i=1}^{n} \min_{j=1}^{k} d(u_i, v_j) \longrightarrow (v_1, \cdots, v_k) = \underset{(v_1,\cdots,v_k)}{\operatorname{argmin}} L$

Solution: soft "L"

1' $\max(\lambda_1, \lambda_2, \cdots, \lambda_n) = \lim_{K\to\infty} \frac{1}{K} \cdot \ln\left(\sum_{i=1}^{n} e^{\lambda_i \cdot K}\right) \approx \frac{1}{K} \ln\left(\sum_{i=1}^{n} e^{\lambda_i K}\right)$

↑此处有一个漂亮的证明

todo 2

2' $\min(\lambda_1, \lambda_2, \cdots, \lambda_n) = -\max(-\lambda_1, -\lambda_2, \cdots, -\lambda_n)$

3' $L \approx -\frac{1}{K} \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{k} e^{-K \cdot d(u_i, v_j)}\right) = -\frac{1}{K} \sum_{i=1}^{n} \ln Z_i$ （近似的 loss 全局光滑可导）

4' $\frac{\partial L}{\partial v_j} \approx -\frac{1}{K} \cdot \underbrace{\frac{e^{-K \cdot d(u_i, v_j)}}{\sum_{j=1}^{k} e^{-K \cdot d(u_i, v_j)}}}_{\text{let it be } C_{ij} = \operatorname{softmax}_j(-K \cdot d(u_i, v_j))} \cdot \frac{\partial d(u_i, v_j)}{\partial v_j} \overset{令}{=} 0$，即可迭代求解.

① 使用欧氏距离： $d(u_i, v_j) = \|u_i - v_j\|^2$

$\Rightarrow \frac{\partial d(u_i, v_j)}{\partial v_j} = 2(v_j - u_i)$

$\therefore 0 = 2 \sum_{i=1}^{n} C_{ij}^{(r)} \left(v_j^{(r+1)} - u_i\right) \longrightarrow v_j^{(r+1)} = \frac{\sum_{j=1}^{n} C_{ij}^{(r)} \cdot u_i}{\sum_{j=1}^{n} C_{ij}^{(r)}}$

② 使用内积相似度： $d(u_i, v_j) = -\langle u_i, v_j \rangle$，but d don't have low boundary!

② Gaussian Mixed Model (GMM) as clustering algorithm.（使用概率分布来描述类别）

Given $x_1, x_2, \cdots, x_n$, find a $p(x)$ satisfied $x_i$.

$\Rightarrow p(x) = \sum_{j=1}^{k} P(j) \cdot P(x|j)$, where j represents class. $P(j) = \pi_j$（常数分布）

with $N(x; \mu_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2} (\det \Sigma_j)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\right)$

we will obtain:

$p(x) = \sum_{j=1}^{k} \underset{\downarrow\ \pi_j}{p(j)} \times \underset{\downarrow\ N(x;\mu_j,\Sigma_j)}{p(x|j)} = \sum_{j=1}^{k} \pi_j N(x; \mu_j, \Sigma_j)$

Solution to determine : $\pi_j, \mu_j, \Sigma_j$ via EM algorithm.



$$P(j|x) = \frac{P(x|j)\cdot P(j)}{P(x)} = \frac{\pi_j \cdot N(x_i; \mu_j, \Sigma_j)}{\sum_{j=1}^{k} \pi_j \cdot N(x_i; \mu_j, \Sigma_j)}$$

① For $\mu_j = \int P(x|j)\, x \cdot dx = \int P(x) \cdot \frac{P(j|x)}{P(j)} x \cdot dx = E\left[\frac{P(j|x)}{P(j)} X\right] = \frac{1}{n}\sum_{i=1}^{n} \frac{P(j|x_i)}{P(j)} x_i = \frac{1}{\pi_j n}\sum_{i=1}^{n} P(j|x_i)\cdot x_i$

②

Likewise, For $\Sigma_j = \frac{1}{\pi_j n}\sum_{i=1}^{n} P(j|x_i)(x_i - \mu_j)(x_i - \mu_j)^T$

③ $\pi_j = P(j) = \int P(j|x)\cdot P(x)\cdot dx = E[P(j|X)] = \frac{1}{n}\sum_{i=1}^{n} P(j|x_i)$

EM: 1' $P(j|x_i) \leftarrow \frac{\pi_j N(x_i; \mu_j, \Sigma_j)}{\sum_{j=1}^{k} \pi_j N(x_i; \mu_j, \Sigma_j)}$

2' $\mu_j \leftarrow \frac{1}{\sum_{j=1}^{k} P(j|x_i)} \sum_{i=1}^{n} P(j|x_i)\cdot x_i$

3' $\Sigma_j \leftarrow \frac{1}{\sum_{j=1}^{n} P(j|x_i)} \sum_{i=1}^{n} P(j|x_i)\cdot (x_i - \mu_j)(x_i - \mu_j)^T$

4' $\pi_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} P(j|x_i)$