

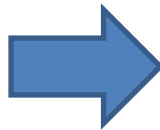
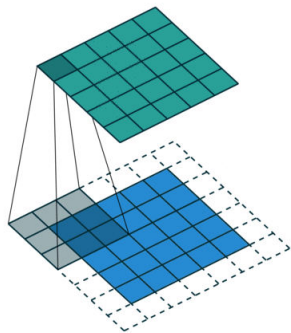


Deep Learning Italia

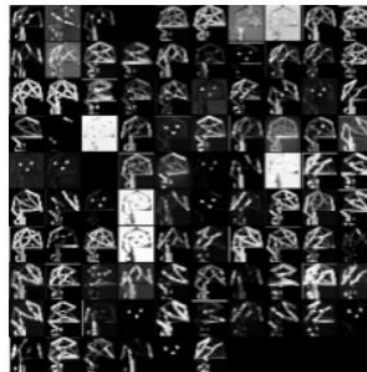
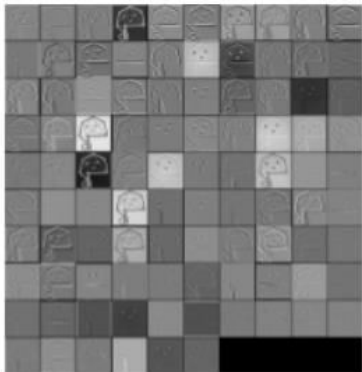
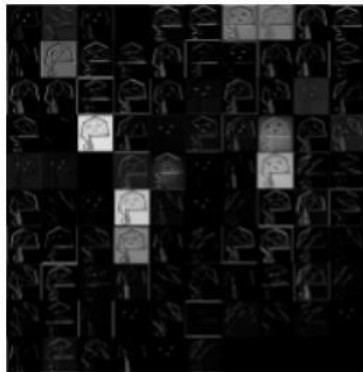
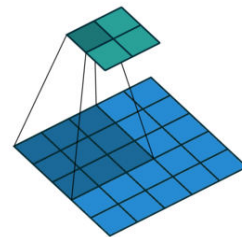
Speaker: Matteo Alberti Deep Learning
Consultant @ Tedge | Community
Machine Learning @
DeepLearningItalia



Convoluzione

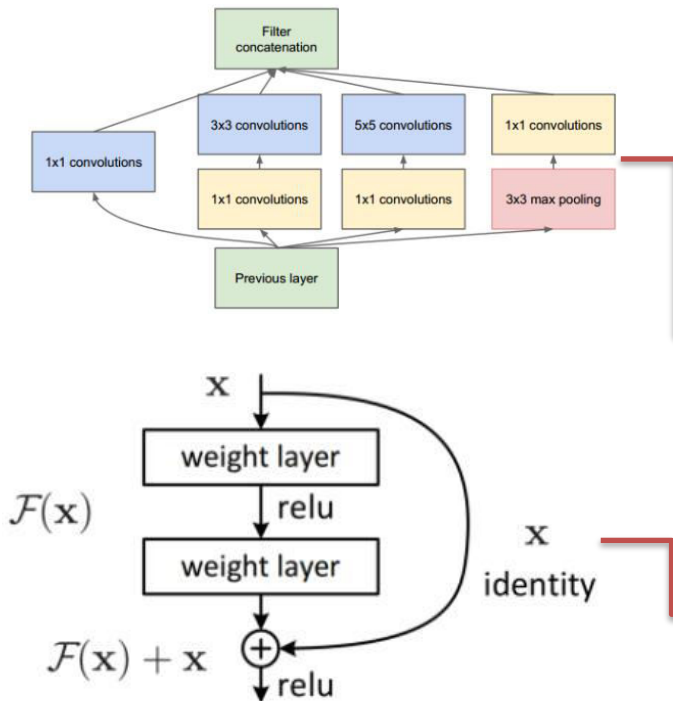
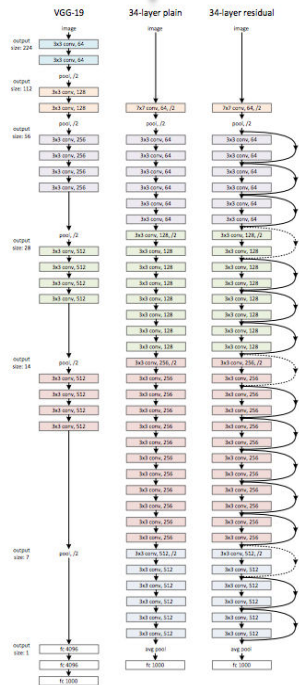


Pooling





Deep Models



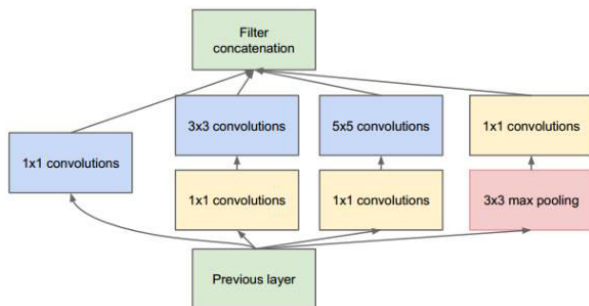
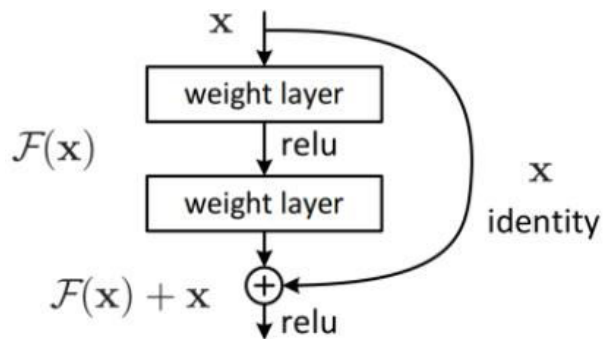
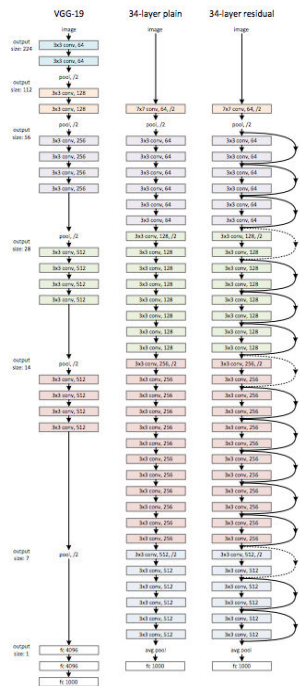
Rete VGG	A, A-LRN con 11 strati	B con 13 strati	C con 16 strati	D con 16 strati	E con 19 strati
Numero di parametri in milioni (M)	133M	133M	134M	138M	144M

❖ Inception Layer

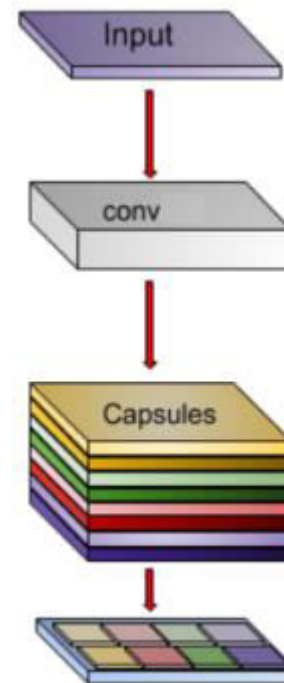
❖ Residual Block
❖ Residual Learning



Deep Models

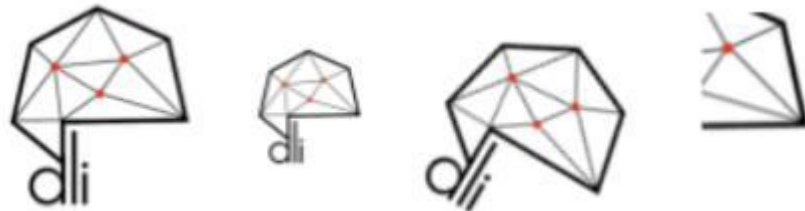
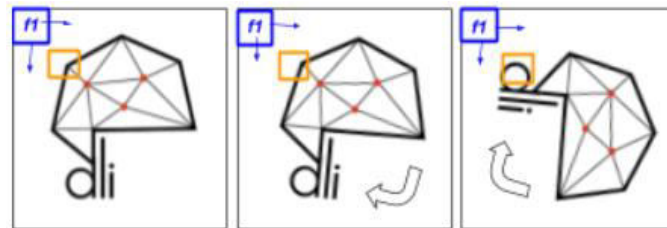
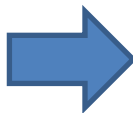
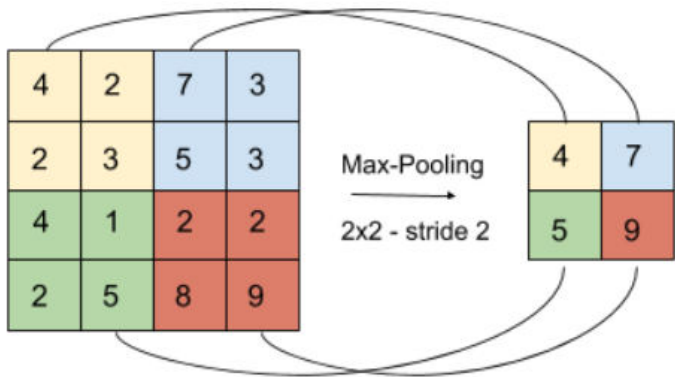


Shallow Models





Il problema fondamentale: *Pooling*



Data Augmentation

Tendiamo a *Memorizzare* piuttosto
che *Generalizzare*



Vi sono tre fondamentali chiavi di differenza:

❖ Da Neuroni a Capsule



Rectangle
x=20 y=30 angle=16°

Triangle
x=24 y=25 angle=-65°

Una capsula non è altro che un gruppo di neuroni. Invece di ricevere in input e produrre scalari riceve un vettore e restituisce un vettore. Il vettore risultante ingloba sia l'output classico sia informazioni spaziali della feature estratta.



Vi sono tre fondamentali chiavi di differenza:

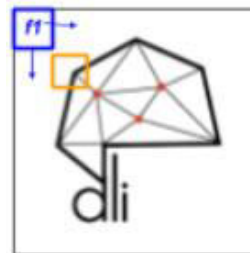
❖ Da Neuroni a Capsule

Una capsula non è altro che un gruppo di neuroni. Invece di ricevere in input e produrre scalari riceve un vettore e restituisce un vettore. Il vettore risultante ingloba sia l'output classico sia informazioni spaziali della feature estratta.

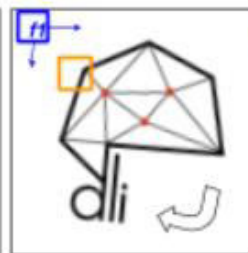
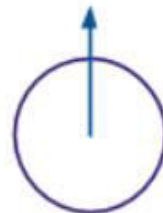


Rectangle
x=20 y=30 angle=16°

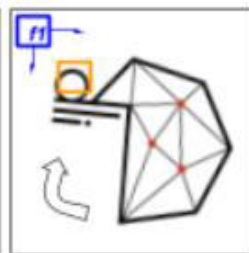
Triangle
x=24 y=25 angle=-65°



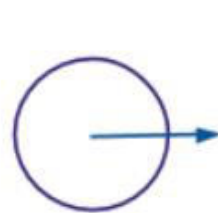
A



B



C





Vi sono tre fondamentali chiavi di differenza:

- ❖ Da Neuroni a *Capsule*
 - ❖ Nuova funzione di *attivazione*

Squash-Function

$$\mathbf{V}_j = \underbrace{\frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2}}_{\text{additional "squashing"}} \underbrace{\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}}_{\text{unit scaling}}$$

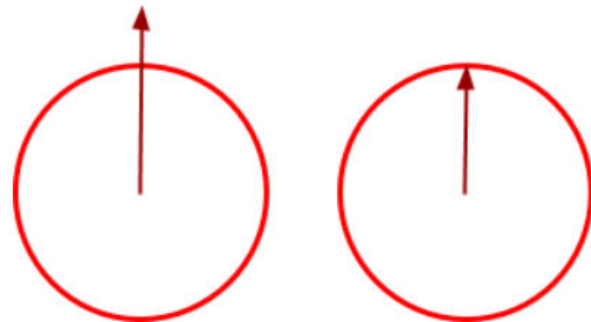


Vi sono tre fondamentali chiavi di differenza:

- ❖ Da Neuroni a *Capsule*
 - ❖ Nuova funzione di *attivazione*

Squash-Function

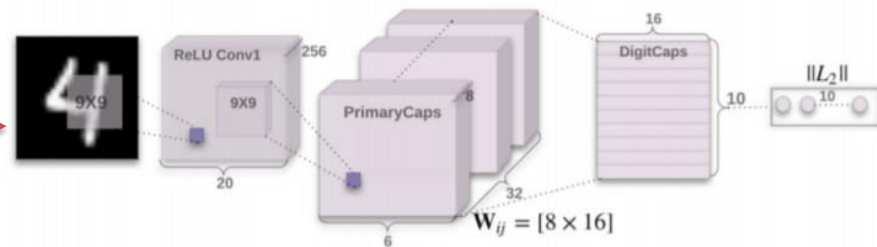
$$\mathbf{V}_j = \underbrace{\frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2}}_{\text{additional "squashing"}} \underbrace{\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}}_{\text{unit scaling}}$$





Vi sono tre fondamentali chiavi di differenza:

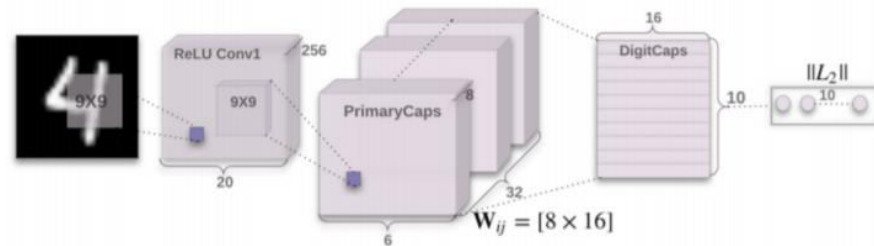
- ❖ Da Neuroni a *Capsule*
 - ❖ Nuova funzione di attivazione
- ❖ Da Deep a **Shallow**





Vi sono tre fondamentali chiavi di differenza:

- ❖ Da Neuroni a *Capsule*
 - ❖ Nuova funzione di attivazione
- ❖ Da Deep a **Shallow**



Convolution (x 2)
Reshape feature maps to 32 groups of 8 feature maps each of size 6 by 6
(**6x6x32=1152 primary capsules**)
Dynamic Routing
Digit Caps (higher level capsules of size 16x1)
Compute Loss
Backpropagate



Vi sono tre fondamentali chiavi di differenza:

- ❖ Da Neuroni a *Capsule*
 - ❖ Nuova funzione di *attivazione*
- ❖ Da Deep a *Shallow*
- ❖ Da Pooling a ***Dynamic Routing***

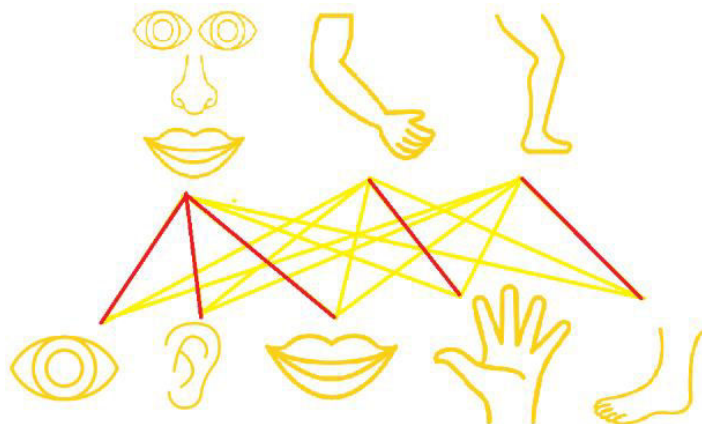
$$\mathbf{s}_j = \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}, \quad \hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij} \mathbf{u}_i \quad c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

Procedure 1 Routing algorithm.

```

1: procedure ROUTING( $\hat{\mathbf{u}}_{j|i}, r, l$ )
2:   for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow 0$ .
3:   for  $r$  iterations do
4:     for all capsule  $i$  in layer  $l$ :  $\mathbf{c}_i \leftarrow \text{softmax}(\mathbf{b}_i)$  ▷ softmax computes Eq. 3
5:     for all capsule  $j$  in layer  $(l + 1)$ :  $\mathbf{s}_j \leftarrow \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$ 
6:     for all capsule  $j$  in layer  $(l + 1)$ :  $\mathbf{v}_j \leftarrow \text{squash}(\mathbf{s}_j)$  ▷ squash computes Eq. 1
7:     for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$ 
   return  $\mathbf{v}_j$ 

```





Vi sono tre fondamentali chiavi di differenza:

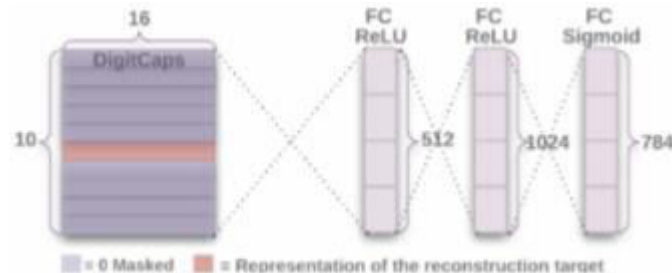
- ❖ Da Neuroni a Capsule
 - ❖ Nuova funzione di attivazione
- ❖ Da Pooling a Dynamic Routing
- ❖ Da Deep a Shallow
 - ❖ **Loss Function**

First Loss Function

$$L_k = T_k \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda (1 - T_k) \max(0, \|\mathbf{v}_k\| - m^-)^2$$



Second Loss Function

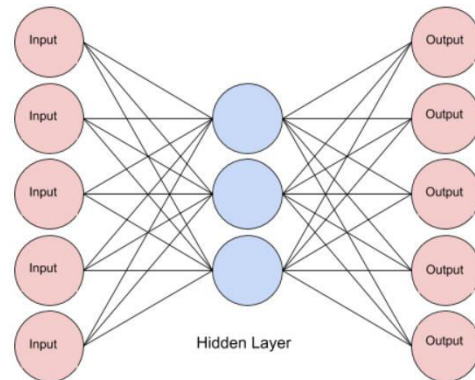




Vi sono tre fondamentali chiavi di differenza:

- ❖ Da Neuroni a Capsule
 - ❖ Nuova funzione di attivazione
- ❖ Da Pooling a Dynamic Routing
- ❖ Da Deep a Shallow
 - ❖ **Loss Function**

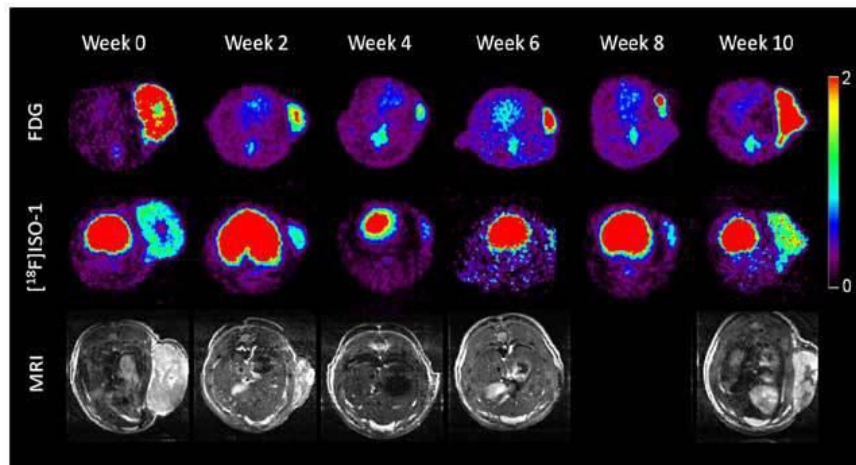
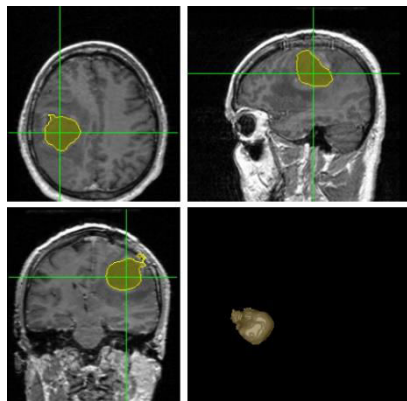
Second Loss Function works as AE



R: (2, 7) L: (2, 7)	R: (6, 0) L: (6, 0)	R: (6, 8) L: (6, 8)	R: (7, 1) L: (7, 1)	*R: (5, 7) L: (5, 0)	*R: (2, 3) L: (4, 3)	R: (2, 8) L: (2, 8)	R: P: (2, 7) L: (2, 8)
4	6	8	7	5	9	2	2
4	6	8	7	5	3	2	2



Use Cases



Quadro generale

- ❖ Forti problemi di segmentazione
- ❖ Classificazione di classi sovrapposte da immagini tridimensionali

Limiti

- ❖ Immagini a dimensionalità ridotta
- ❖ *Routing non ottimizzato e difficile convergenza*