CNN

Arquiteturas e Backpropagation

Prof. Frederico Coelho.

CNN - recaptulando

Tipos de camadas

- Convolucional : Definem os filtros (Aprendizado / BackPropagation)

- Ativação: Neurônios (Relu / Sigmoid / TangH)

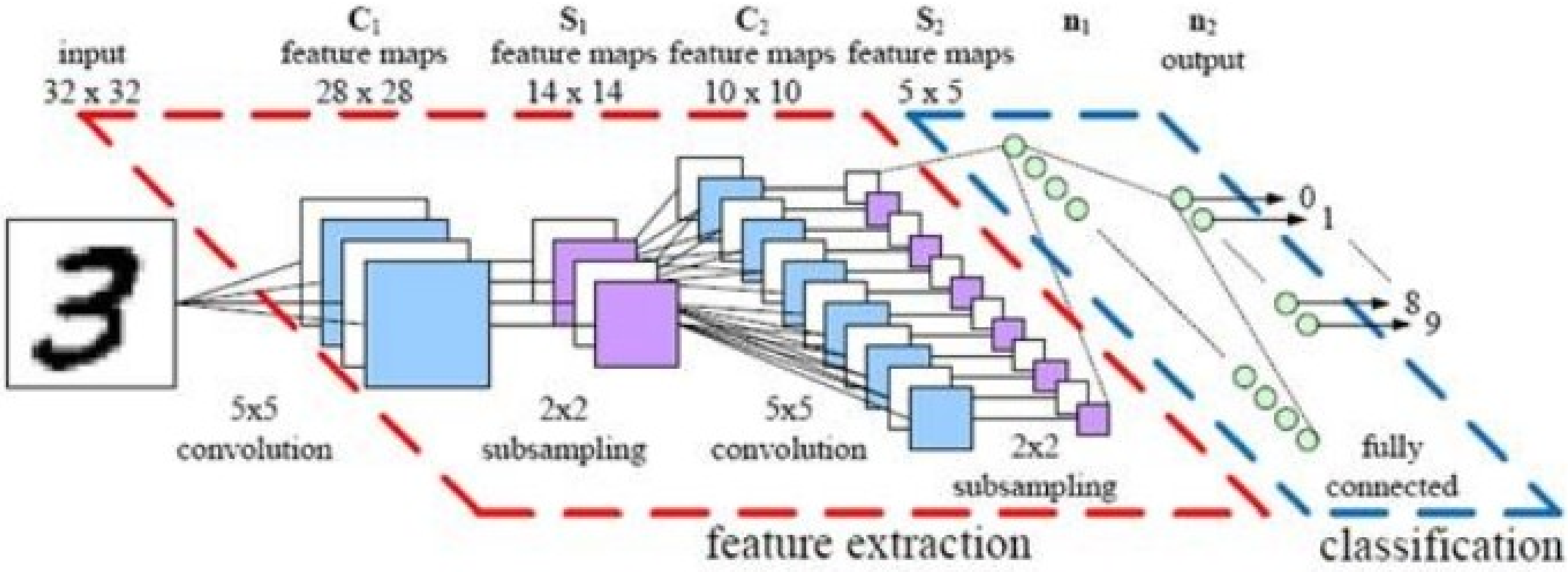
- Pooling : Reduzem as escalas (Max, Median, etc..)

- Fully-Connected (FC): Camada que determina as classes (Classificador)

CNN - recaptulando

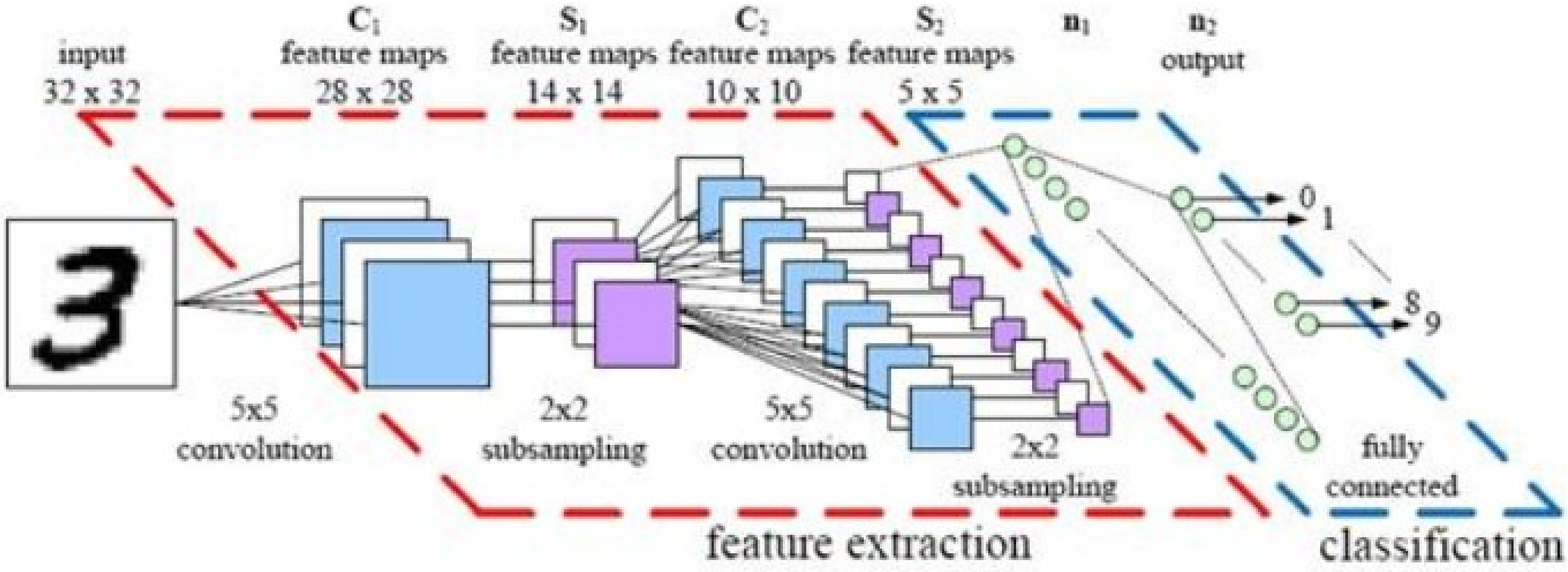
- Compostas de duas grandes etapas:

→Extração de Características pelas Camadas Convolucionais
→Classificação



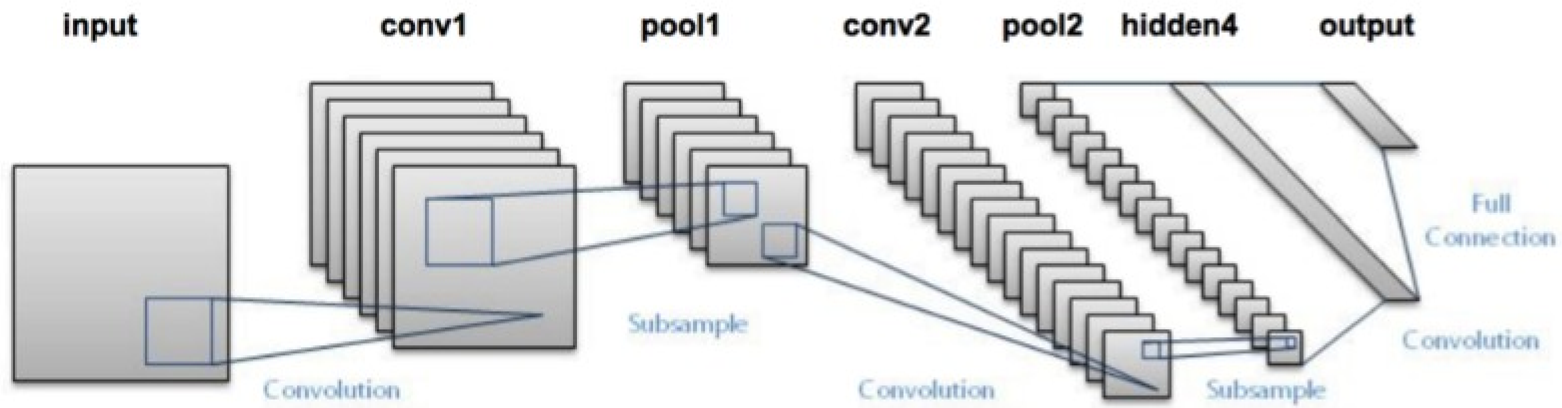
CNN - Arquiteturas

A maneira como estas camadas são conectadas definem as arquiteturas.



CNN - Arquiteturas

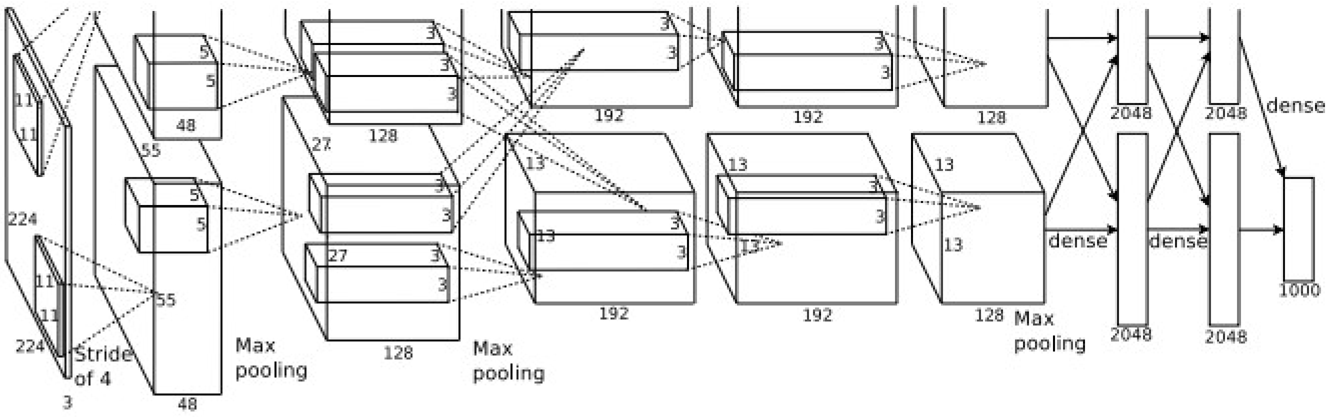
LeNet-5 (1998)



Pioneira – aplicada a bancos de dados de escrita manual para classificação de dígitos.

CNN - Arquiteturas

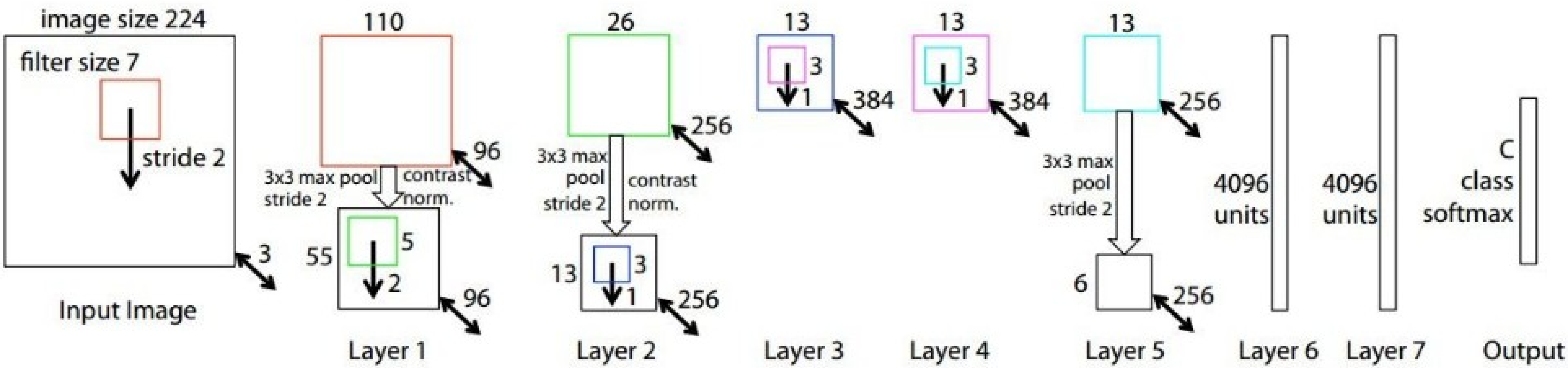
AlexNet (2012)



A rede tinha uma arquitetura muito semelhante à LeNet. Mas era mais profunda, com mais filtros por camada e com camadas convolucionais Empilhadas.

CNN - Arquiteturas

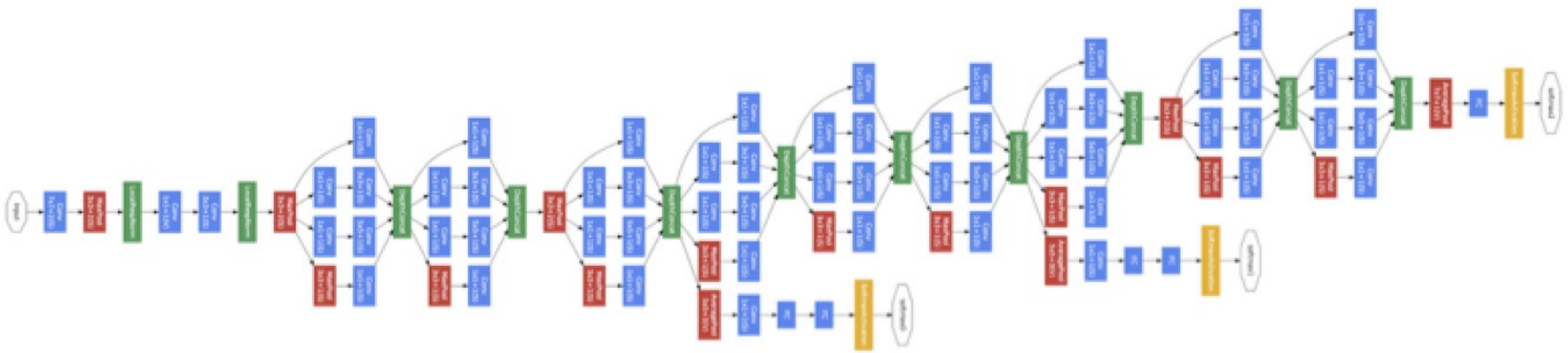
ZFNet (2013)



Foi uma melhoria obtida ajustando os hiperparâmetros do AlexNet, mantendo a mesma estrutura com elementos adicionais de Deep Learning.

CNN - Arquiteturas

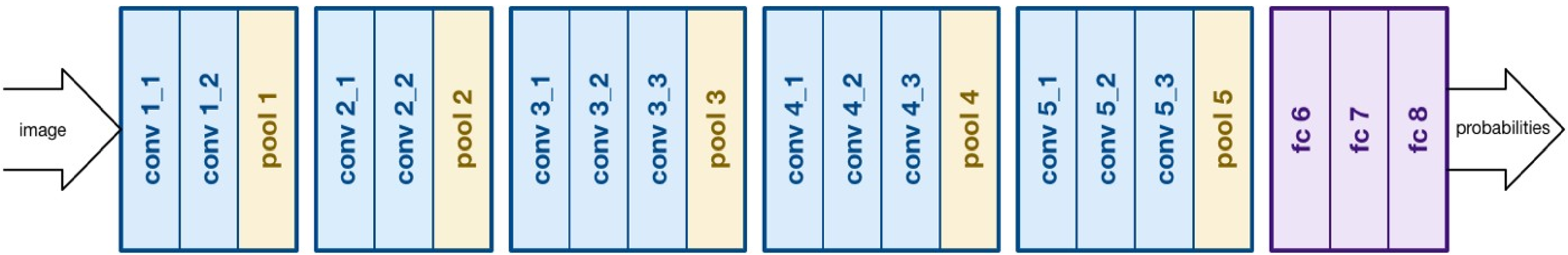
GoogleNet/Inception (2014)



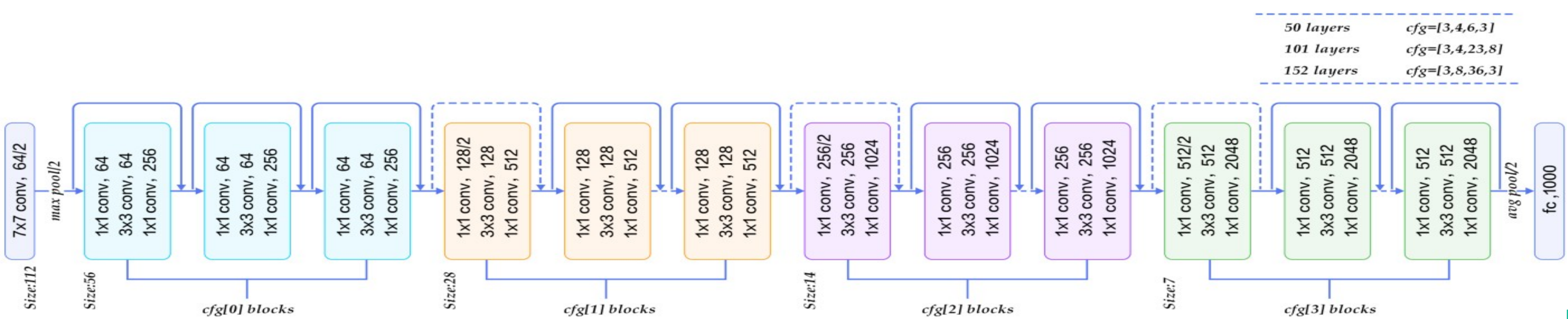
Sua arquitetura consistia em uma CNN com 22 camadas de profundidade, mas reduziu o número de parâmetros de 60 milhões (AlexNet) para 4 milhões.

CNN - Arquiteturas

VGGNet (2014)



ResNet (2015)



CNN - Arquiteturas

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

CNN - Backpropagation

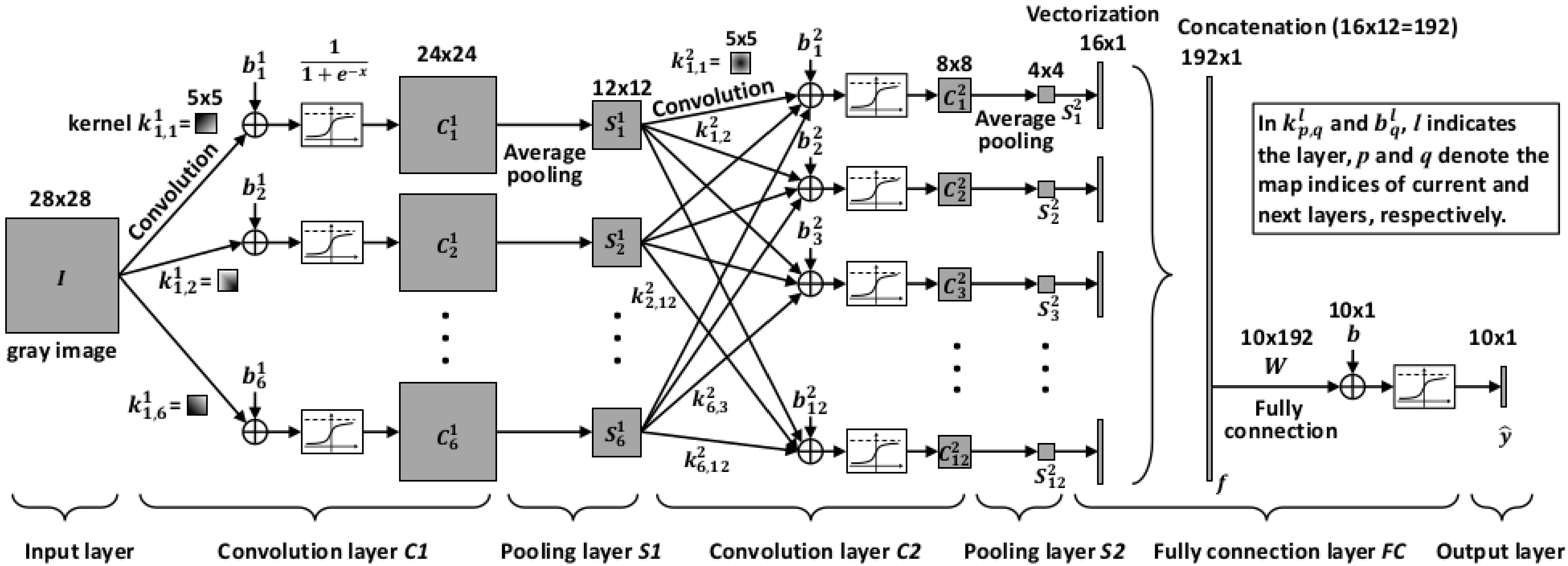
Artigo

Derivation of Backpropagation in Convolutional Neural Network (CNN)

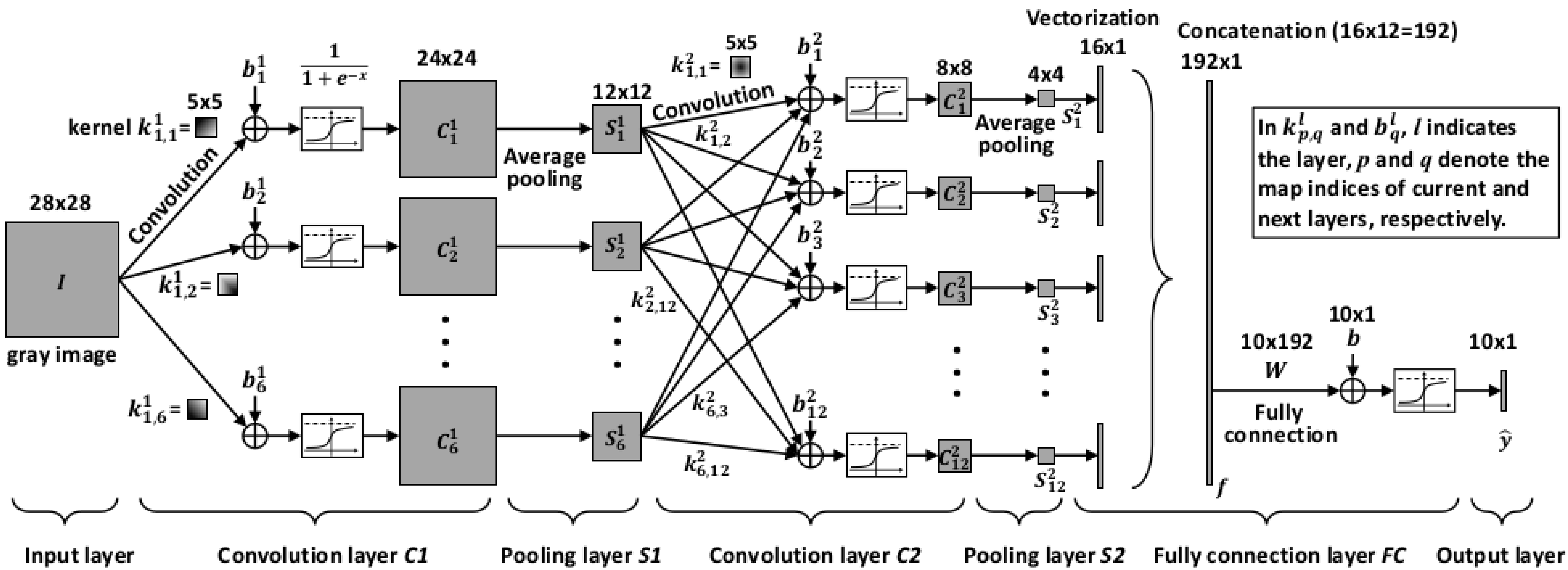
Zhifei Zhang

University of Tennessee, Knoxvill, TN

CNN - Backpropagation



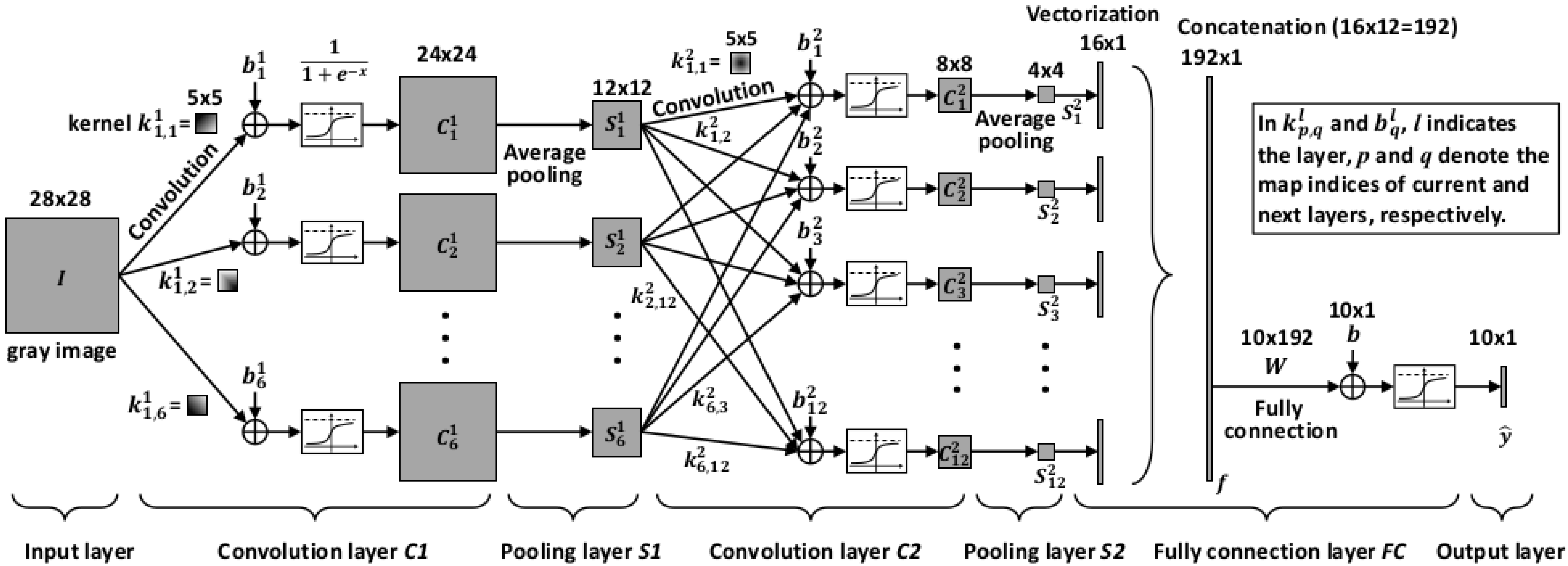
CNN - Backpropagation



Inicialização de parâmetros:

- **C1 layer**, $k^1_{1,p}$ (size 5×5) and b^1_p (size 1×1), $p = 1, 2, \cdots 6$
- **C2 layer**, $k^2_{p,q}$ (size 5×5) and b^2_q (size 1×1), $q = 1, 2, \cdots 12$
- **FC layer**, W (size 10×192) and b (size 10×1)

CNN - Backpropagation



Inicialização de parâmetros:

bias, b^1_p, b^2_q, and b, are initialize to zero

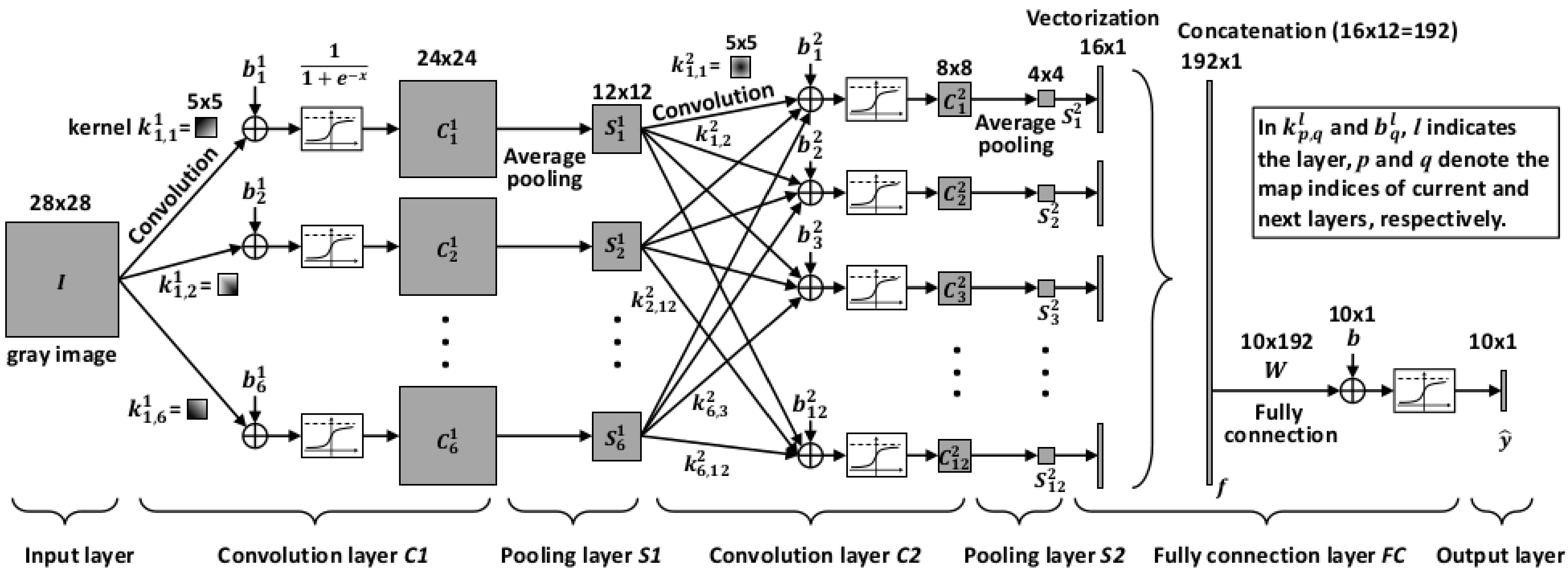
Aleatório com distribuição uniforme

$$k^1_{1,p} \sim U\left(\pm\sqrt{\frac{6}{(1+6)\times 5^2}}\right)$$

$$k^2_{p,q} \sim U\left(\pm\sqrt{\frac{6}{(6+12)\times 5^2}}\right)$$

$$W \sim U\left(\pm\sqrt{\frac{6}{192+10}}\right)$$

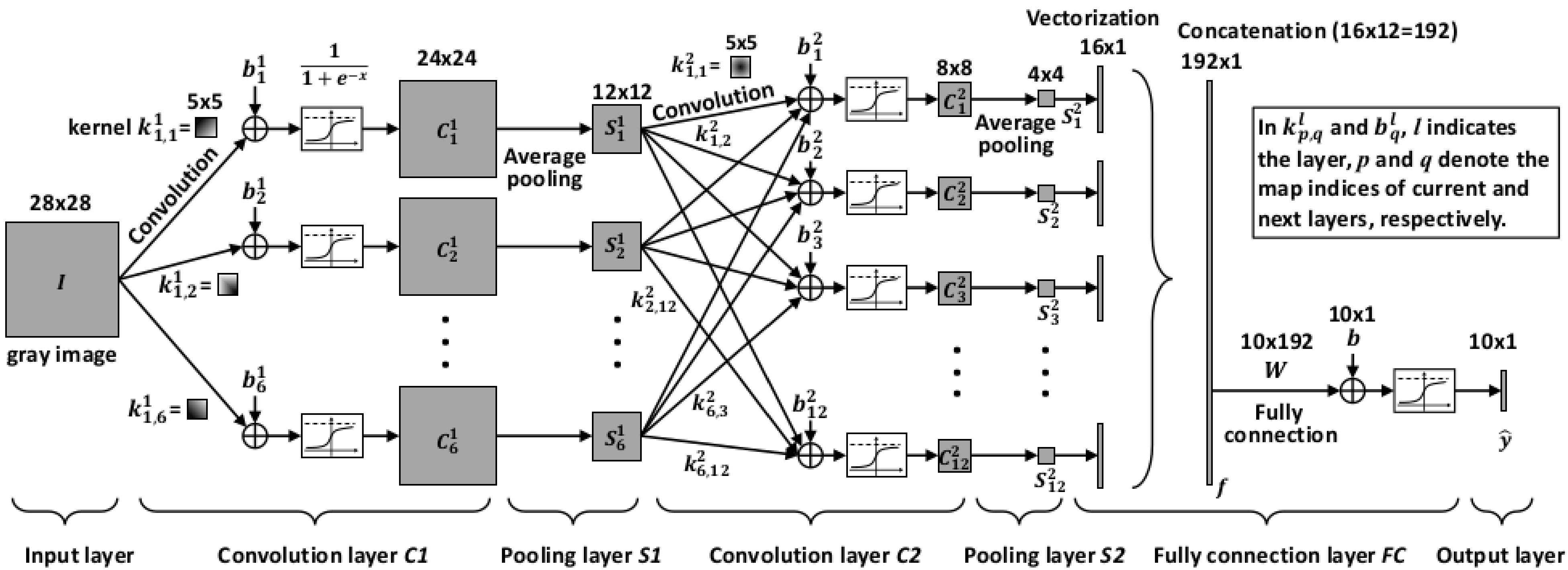
CNN - Backpropagation



$$C_p^1 = \sigma(I * k_{1,p}^1 + b_p^1), \text{ where } \sigma(x) = \frac{1}{1 + \exp^{-x}}$$

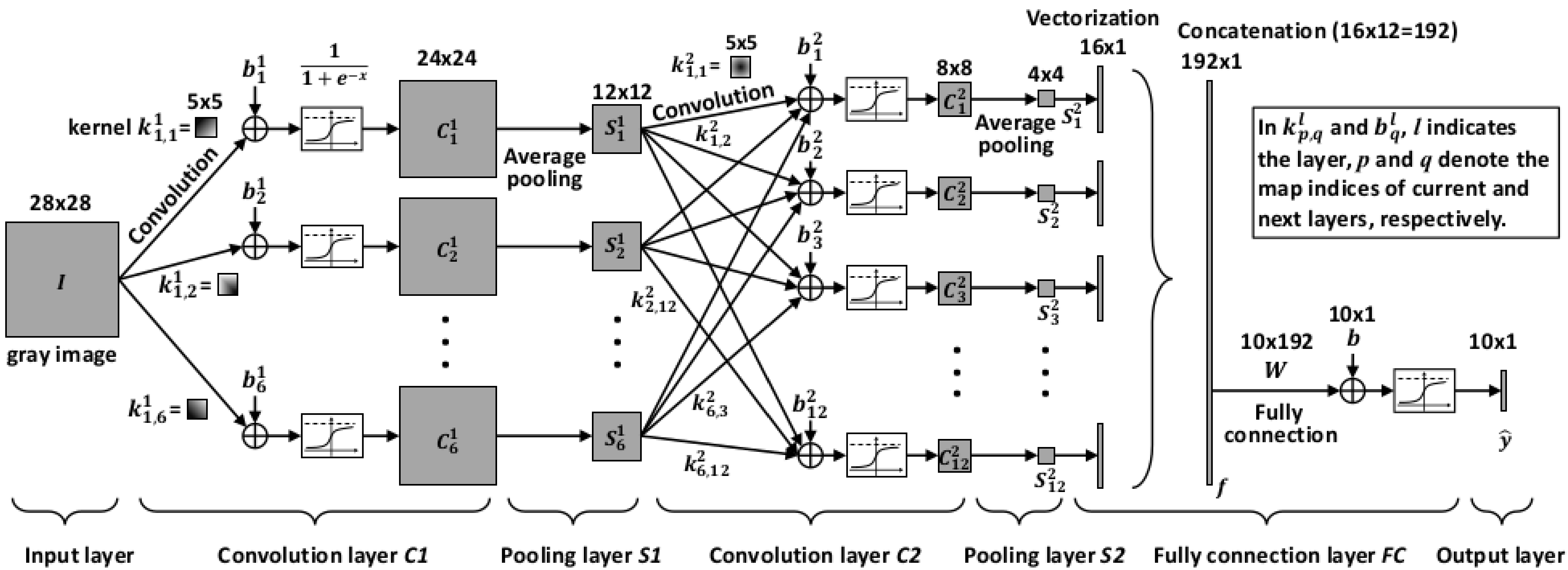
$$C_p^1(i,j) = \sigma\left(\sum_{u=-2}^{2}\sum_{v=-2}^{2} I(i-u, j-v) \cdot k_{1,p}^1(u,v) + b_p^1\right)$$

CNN - Backpropagation



$$S_p^1(i,j) = \frac{1}{4}\sum_{u=0}^{1}\sum_{v=0}^{1} C_p^1(2i-u, 2j-v),\ \ i,j = 1,2,\cdots,12$$

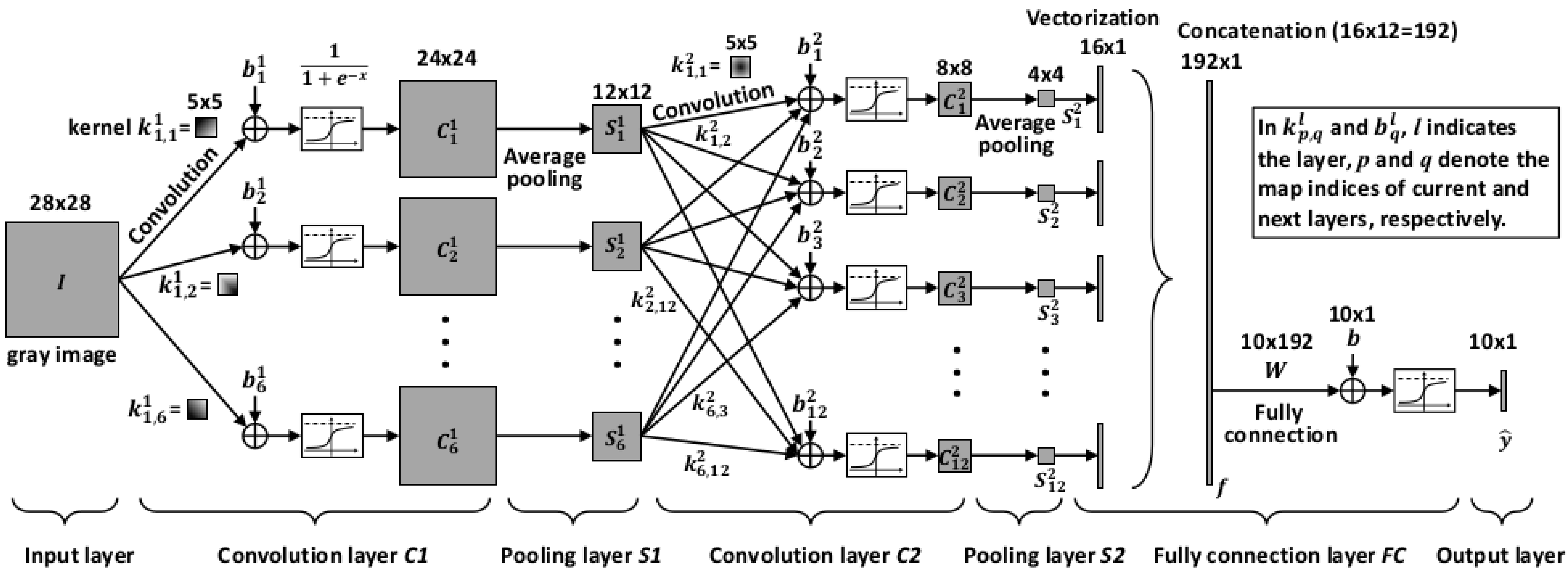
CNN - Backpropagation



$$C_q^2 = \sigma\left(\sum_{p=1}^{6} S_p^1 * k_{p,q}^2 + b_q^2\right)$$

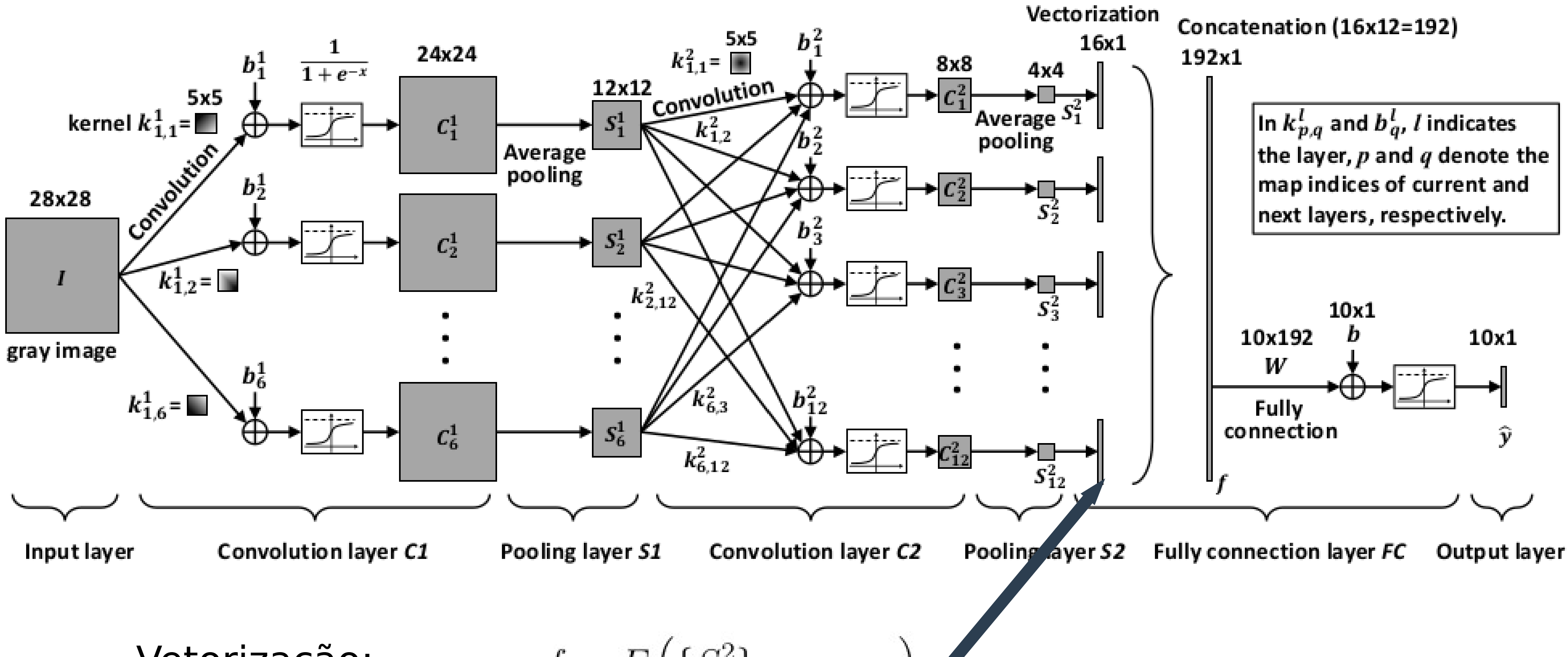
$$C_q^2(i,j) = \sigma\left(\sum_{p=1}^{6}\sum_{u=-2}^{2}\sum_{v=-2}^{2} S_p^1(i-u, j-v)\cdot k_{p,q}^2(u,v) + b_q^2\right)$$

CNN - Backpropagation



$$S_q^2(i,j) = \frac{1}{4}\sum_{u=0}^{1}\sum_{v=0}^{1} C_q^2(2i-u, 2j-v),\ i,j = 1,2,\cdots,4$$

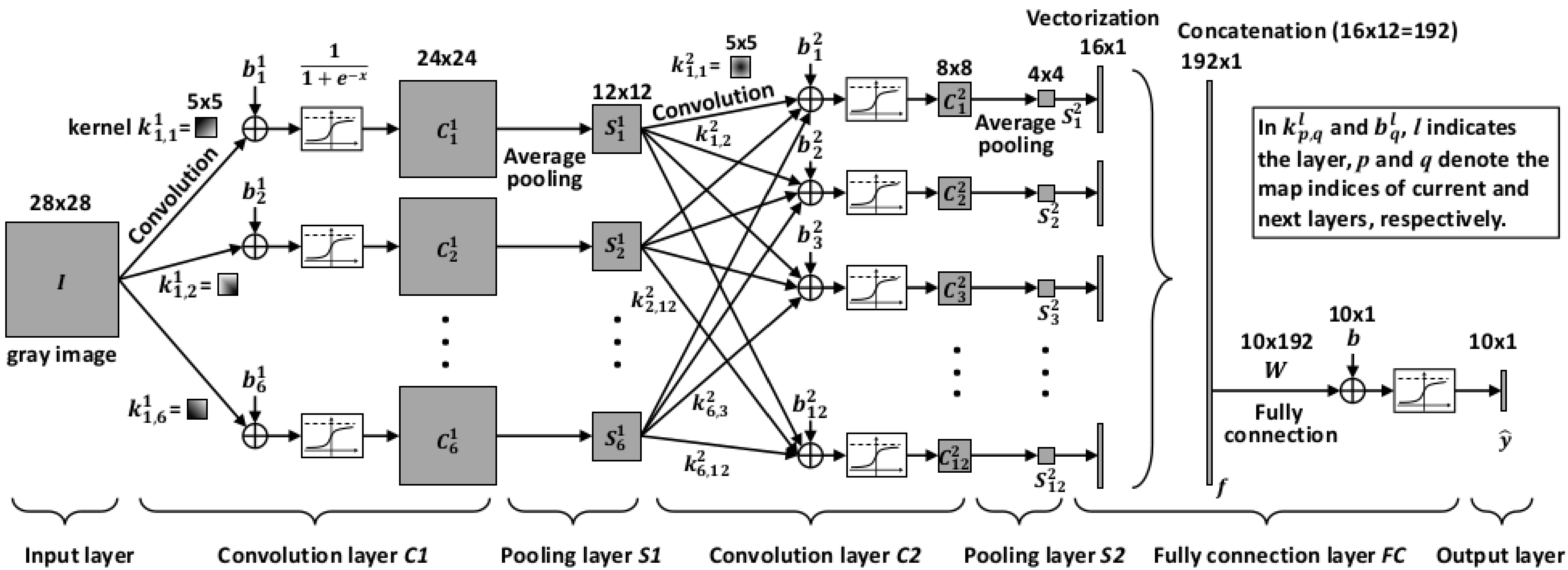
CNN - Backpropagation



Vetorização: $f = F\left(\{S_q^2\}_{q=1,2,\cdots,12}\right)$

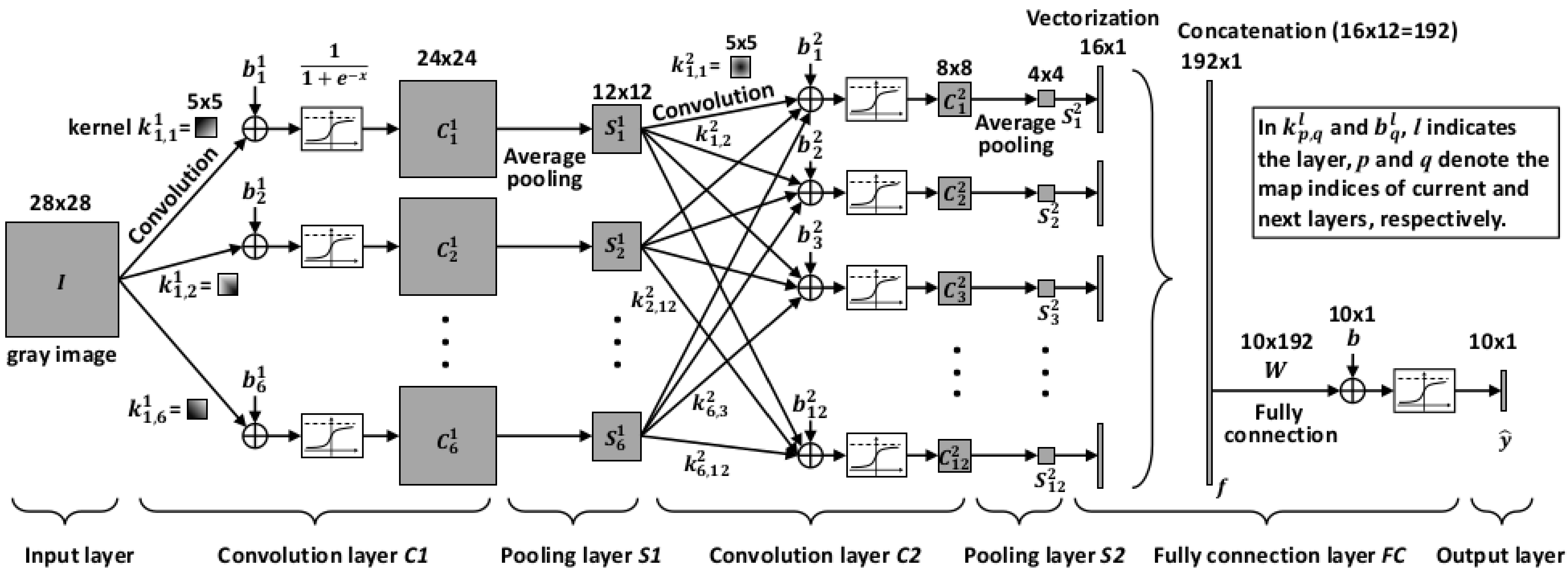
Operação Inversa: $\{S_q^2\}_{q=1,2,\cdots,12} = F^{-1}(f)$

CNN - Backpropagation



$$\hat{y} = \sigma(W \times f + b)$$

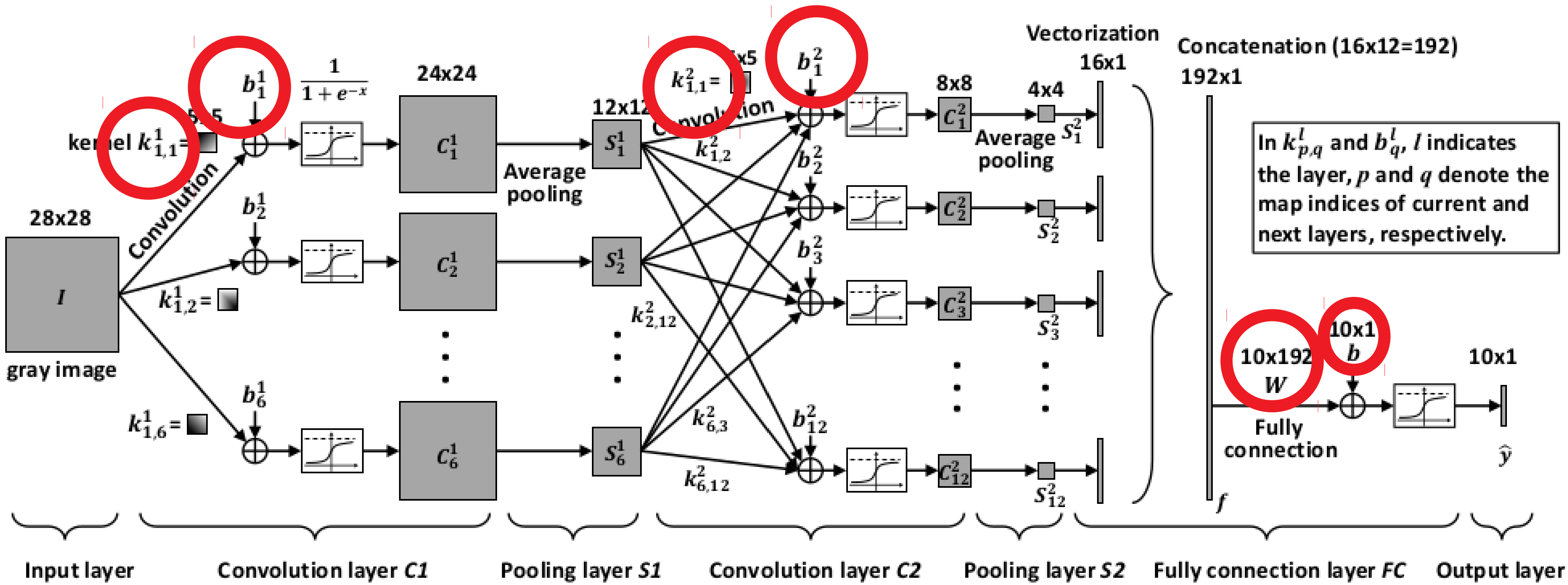
CNN - Backpropagation



Função de custo

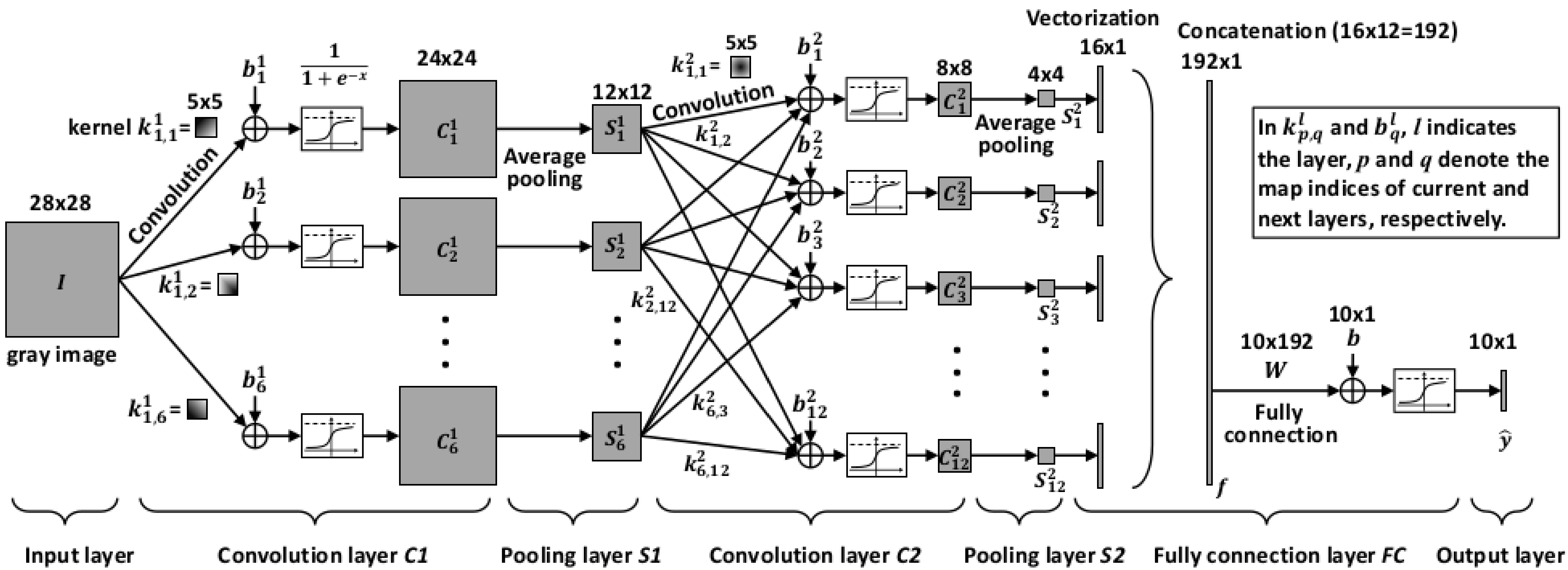
$$L = \frac{1}{2}\sum_{i=1}^{10}\left(\hat{y}(i) - y(i)\right)^2$$

CNN - Backpropagation



Queremos ajustar estes parâmetros para minimizar a função de custo

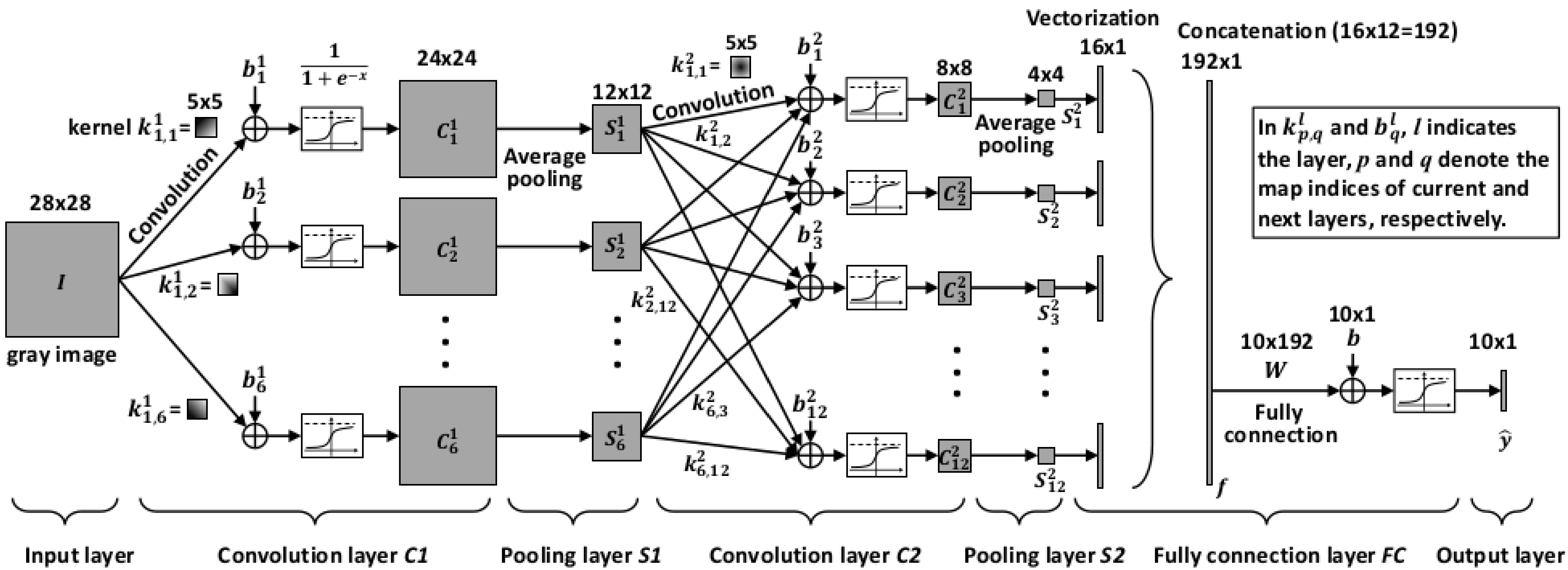
CNN - Backpropagation



$$\Delta W(i,j) = \frac{\partial L}{\partial W(i,j)}$$

$$\Delta W(i,j) = \Delta \hat{y}(i) \cdot f(j)$$
$$\Longrightarrow \Delta W = \Delta \hat{y} \times f^T$$

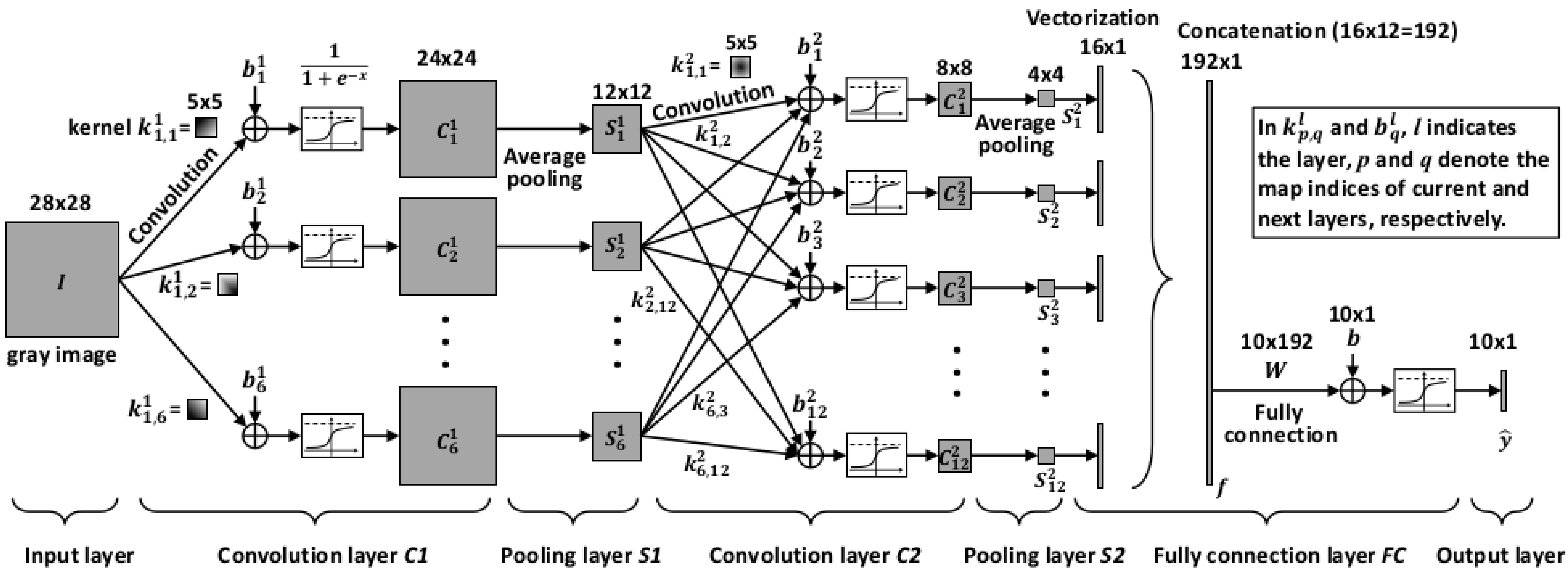
CNN - Backpropagation



$$\Delta b(i) = \frac{\partial L}{\partial b(i)}$$

$$\Delta b = \Delta \hat{y}$$

CNN - Backpropagation



$\Delta k^2_{p,q}$ (size 5×5)

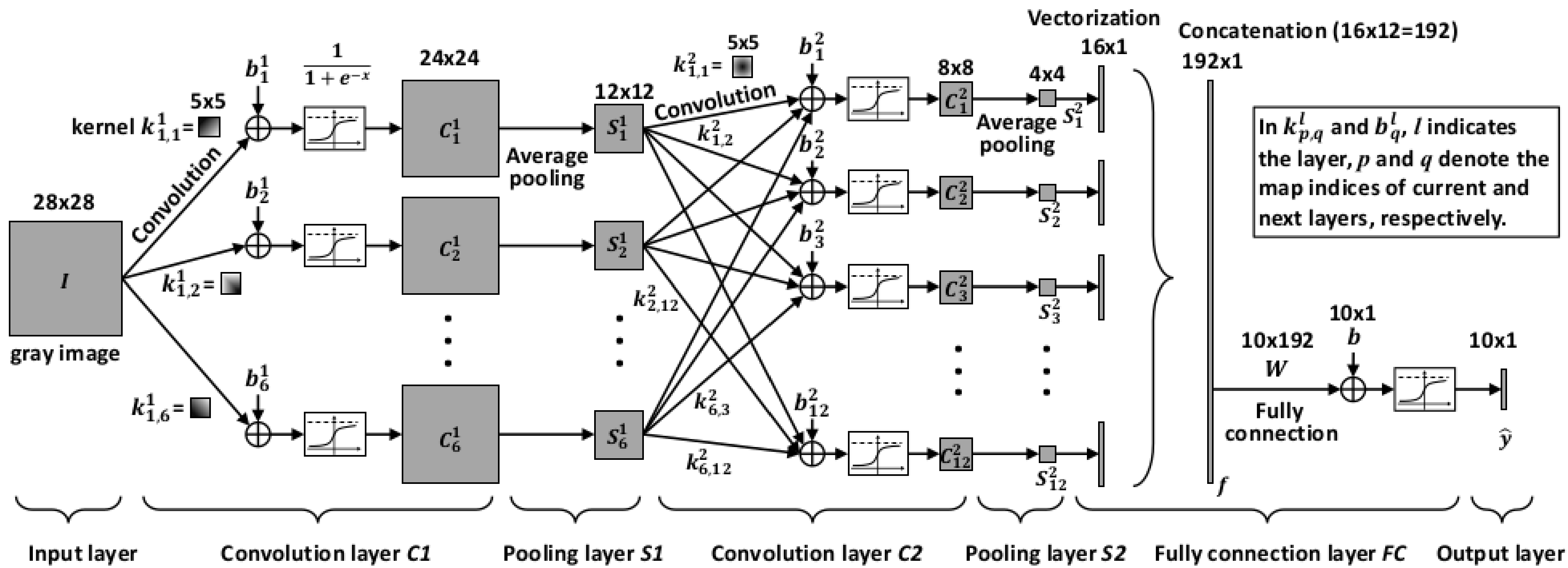
Calculamos $\Delta f(j) = \dfrac{\partial L}{\partial f} \implies \Delta f = W^T \times \Delta \hat{y}$

Reformatamos o vetor $\{\Delta S^2_q\}_{q=1,2,\cdots,12} = F^{-1}(\Delta f)$

Então calculamos (upsampling) $\Delta C^2_q(i,j) = \dfrac{1}{4}\Delta S^2_q\left(\lceil i/2 \rceil, \lceil j/2 \rceil\right),\ i,j = 1,2,\cdots,8$

$\lceil \cdot \rceil$ denotes the ceiling function

CNN - Backpropagation

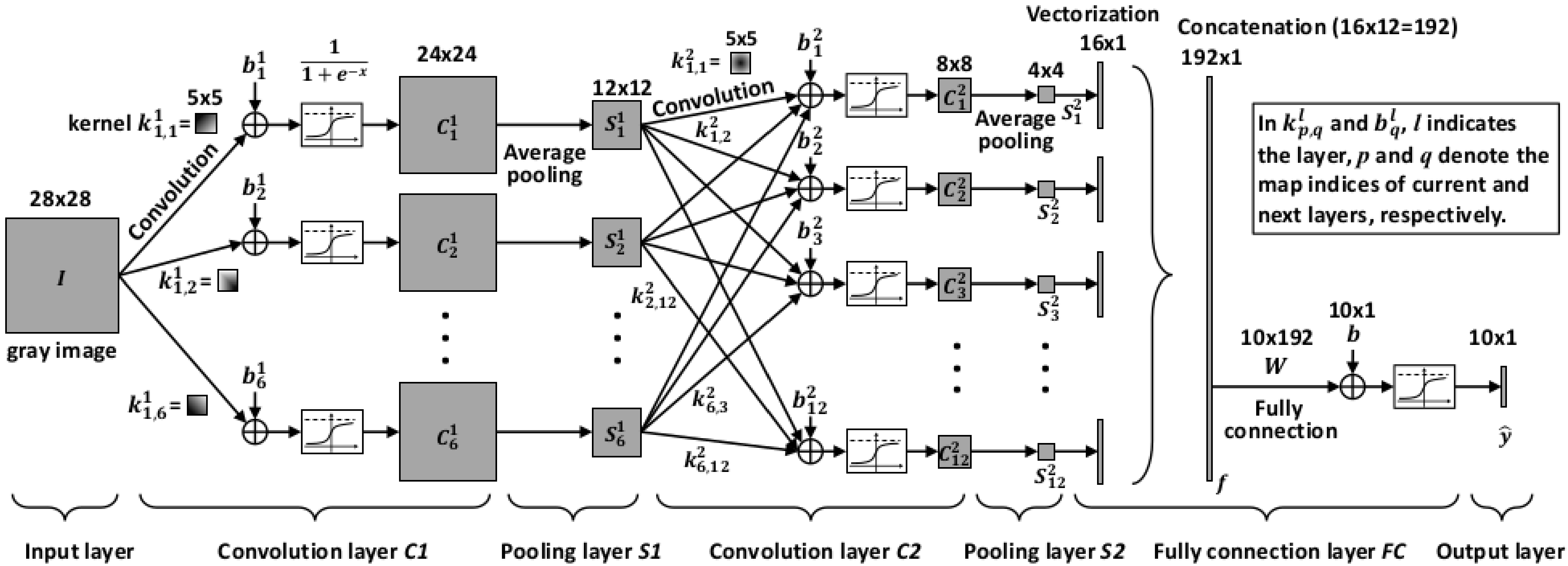


$\Delta k^2_{p,q}$ (size 5×5)

$$\Delta k^2_{p,q}(u,v) = \frac{\partial L}{\partial k^2_{p,q}(u,v)}$$

$$= \sum_{i=1}^{8} \sum_{j=1}^{8} \Delta C^2_q(i,j) \cdot C^2_q(i,j) \left(1 - C^2_q(i,j)\right) \cdot S^1_p(i-u, j-v)$$

CNN - Backpropagation

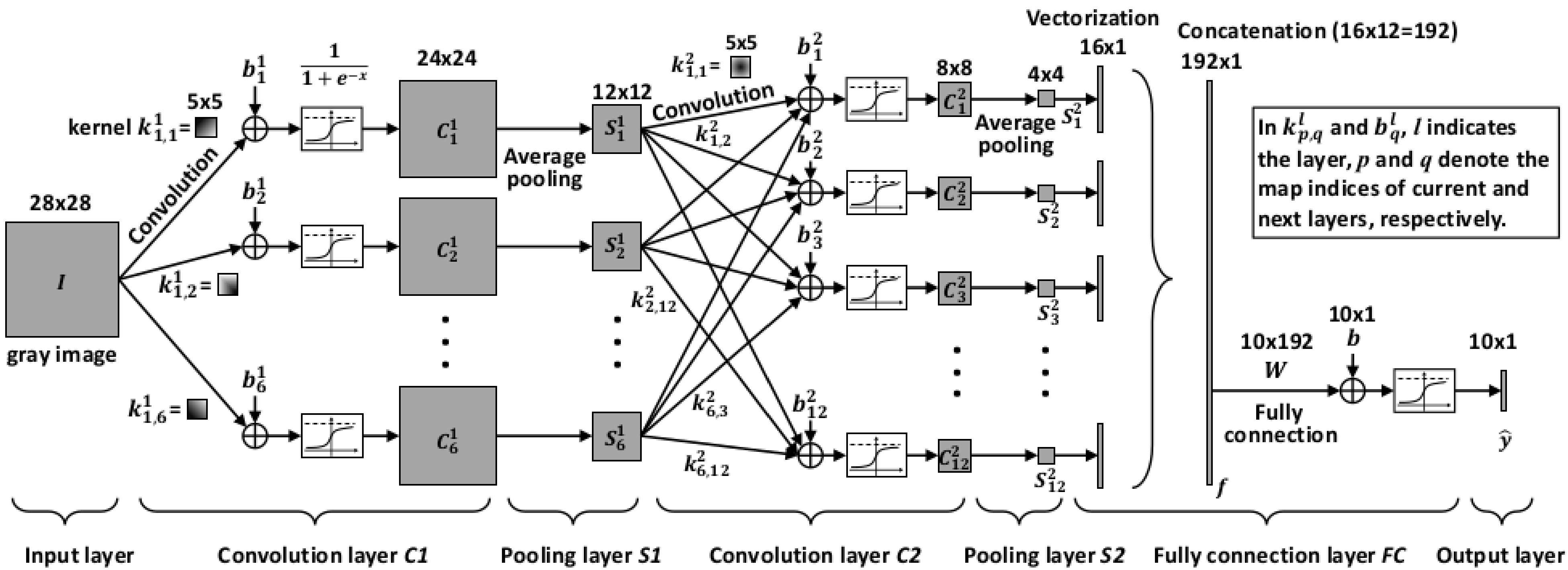


$\Delta k^2_{p,q}$ (size 5×5)

Rotating S^1_p 180 degrees, we get $S^1_{p,rot180}$

$$S^1_{p,rot180}(u-i, v-j) = S^1_p(i-u, j-v)$$

CNN - Backpropagation

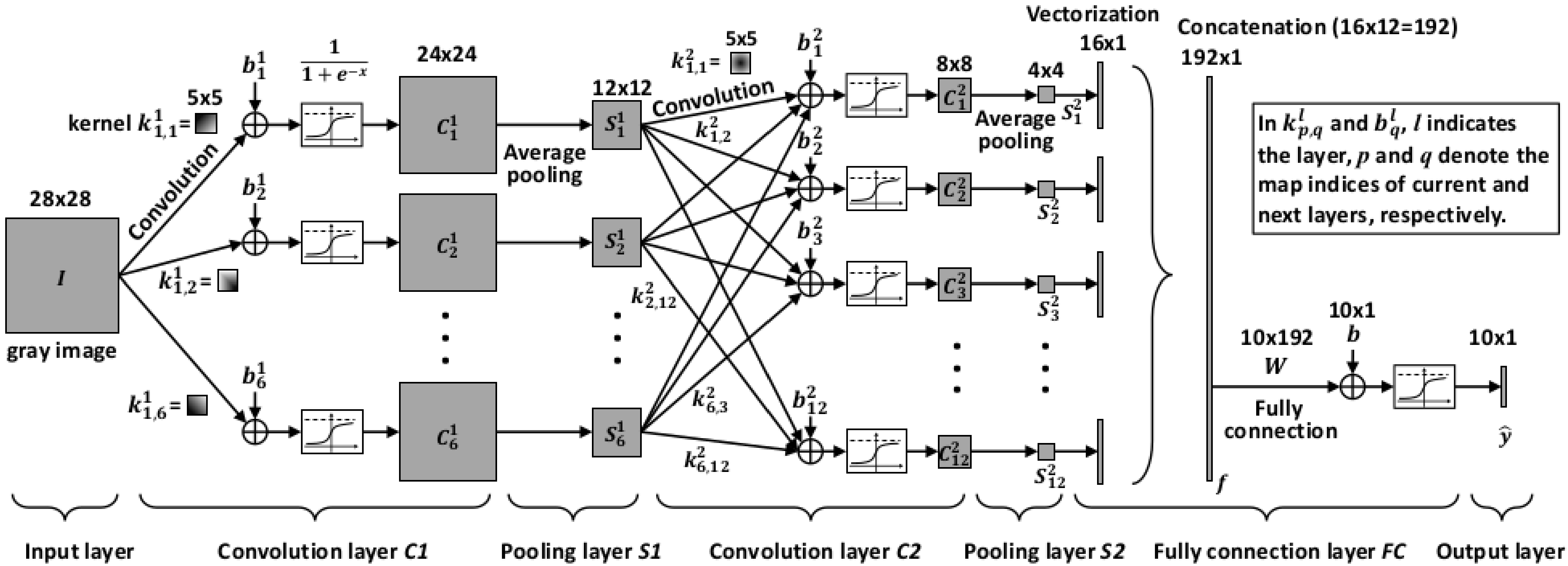


$\Delta k^2_{p,q}$ (size 5×5)

$$\Delta k^2_{p,q}(u,v) = \sum_{i=1}^{8}\sum_{j=1}^{8} S^1_{p,rot180}(u-i, v-j) \cdot \Delta C^2_{q,\sigma}(i,j)$$

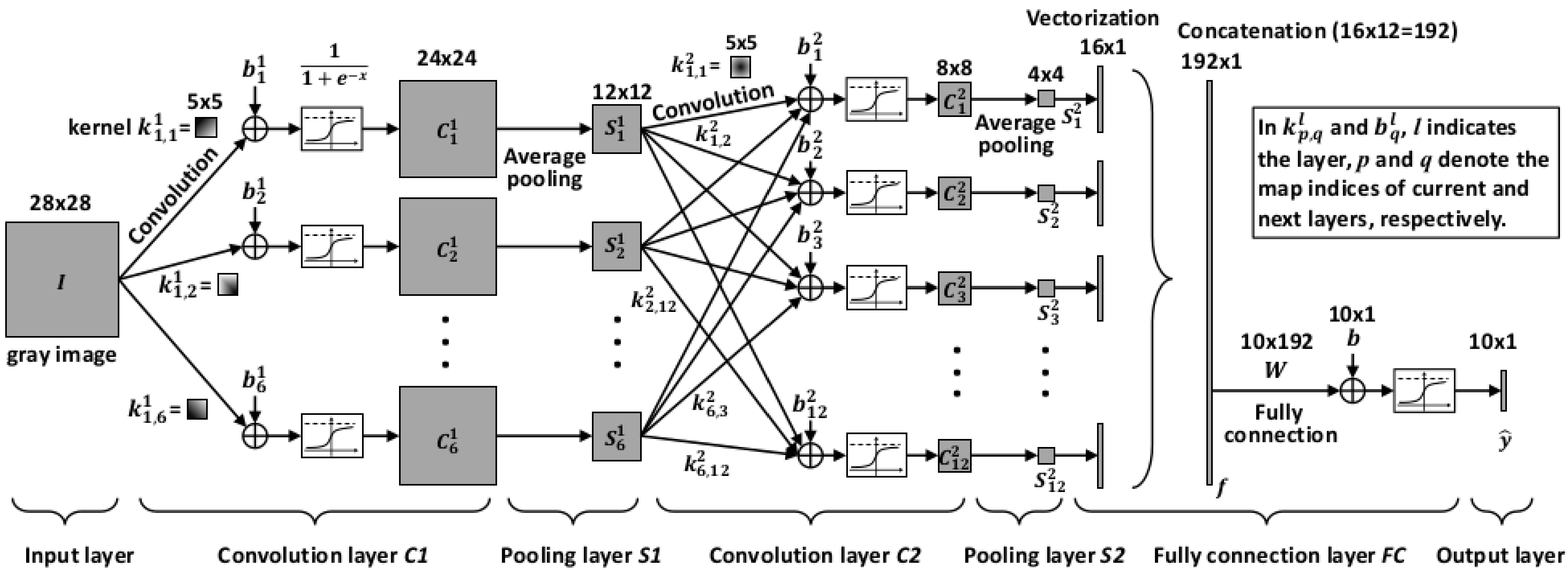
$$\Longrightarrow \Delta k^2_{p,q} = S^1_{p,rot180} * \Delta C^2_{q,\sigma}$$

CNN - Backpropagation



$$\Delta b^2_q = \frac{\partial L}{\partial b^2_q} \quad = \sum_{i=1}^{8}\sum_{j=1}^{8} \Delta C^2_{q,\sigma}(i,j)$$

CNN - Backpropagation

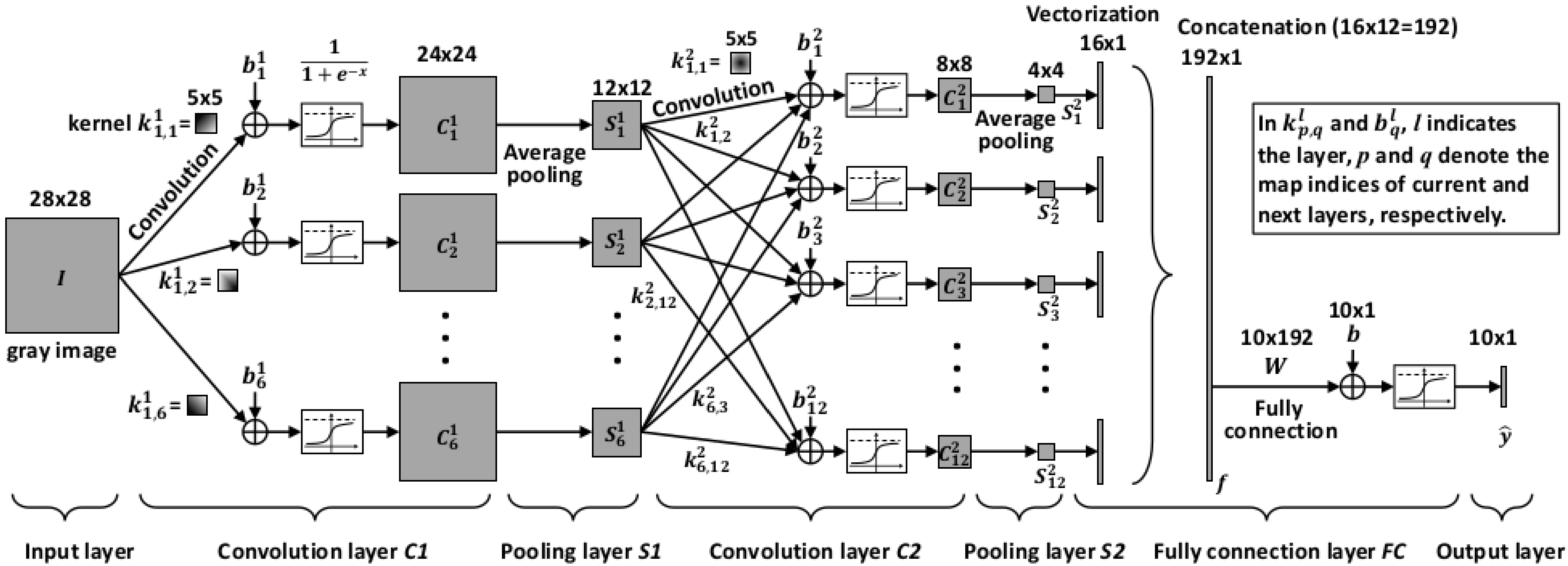


$\Delta k^1_{1,p}$ **(size** 5×5**)**

$$\Delta S^1_p(i,j) = \frac{\partial L}{\partial S^1_p(i,j)} = \sum_{q=1}^{12} \sum_{u=-2}^{2} \sum_{v=-2}^{2} \Delta C^2_{q,\sigma}(i+u, j+v) \cdot k^2_{p,q}(u,v)$$

Rotating $k^2_{p,q}$ 180 degrees, we get $k^2_{p,q,rot180}(-u,-v) = k^2_{p,q}(u,v)$

CNN - Backpropagation

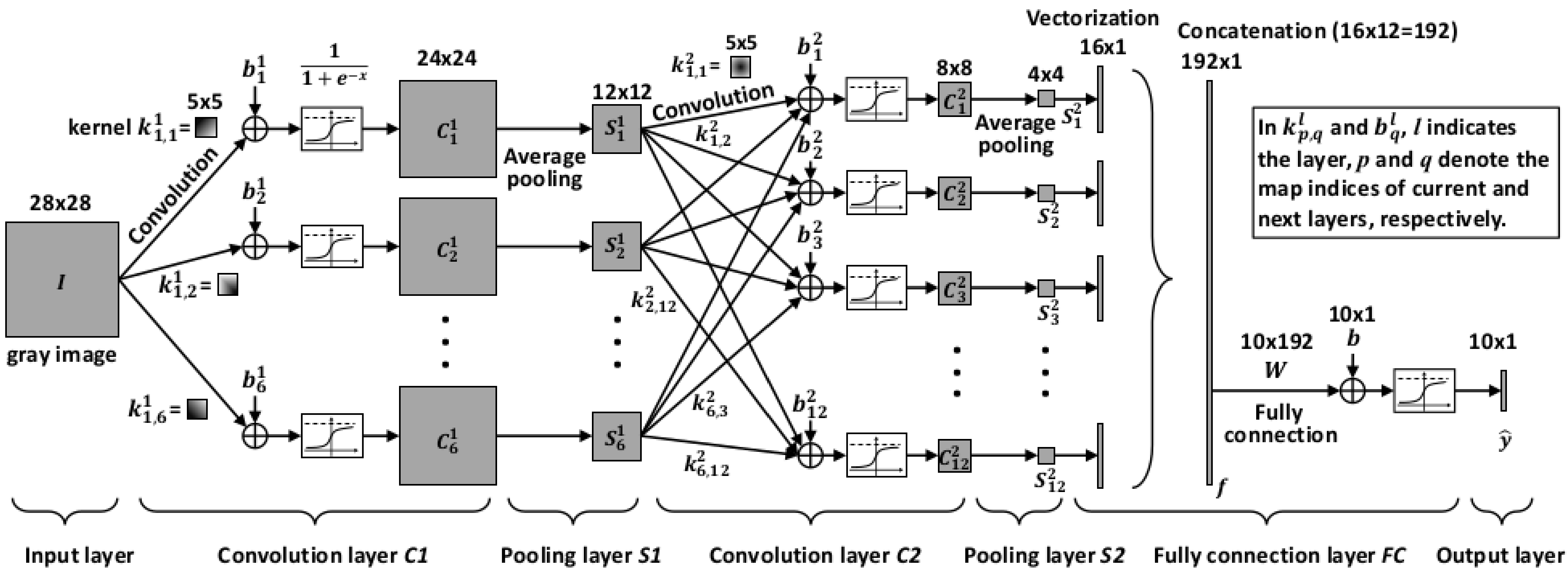


$\Delta k^1_{1,p}$ (size 5×5)

$$\Delta S^1_p(i,j) = \sum_{q=1}^{12} \sum_{u=-2}^{2} \sum_{v=-2}^{2} \Delta C^2_{q,\sigma}(i-(-u), j-(-v)) \cdot k^2_{p,q,rot180}(-u,-v)$$

$$\Longrightarrow \Delta S^1_p = \sum_{q=1}^{12} \Delta C^2_{q,\sigma} * k^2_{p,q,rot180}$$

CNN - Backpropagation

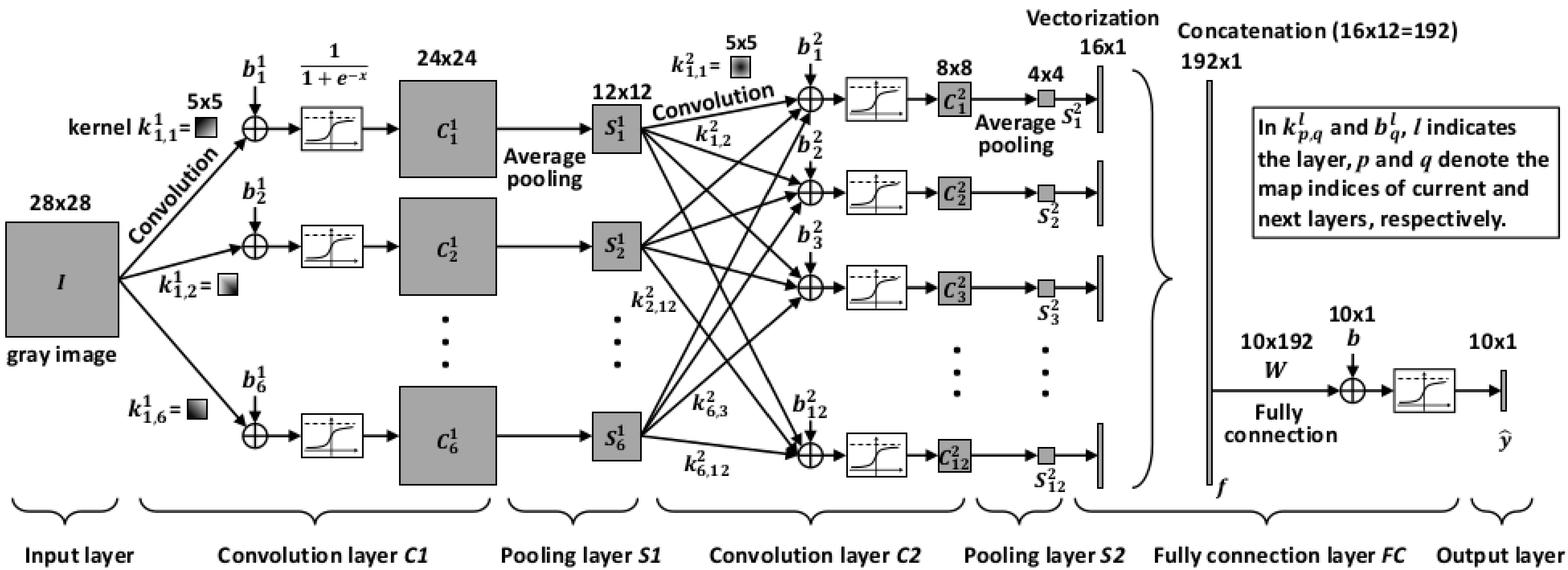


$\Delta k^1_{1,p}$ (size 5×5)

Upsampling

$$\Delta C^1_p(i,j) = \frac{1}{4}\Delta S^1_p\left(\lceil i/2 \rceil, \lceil j/2 \rceil\right),\ i,j = 1,2,\cdots,24$$

CNN - Backpropagation

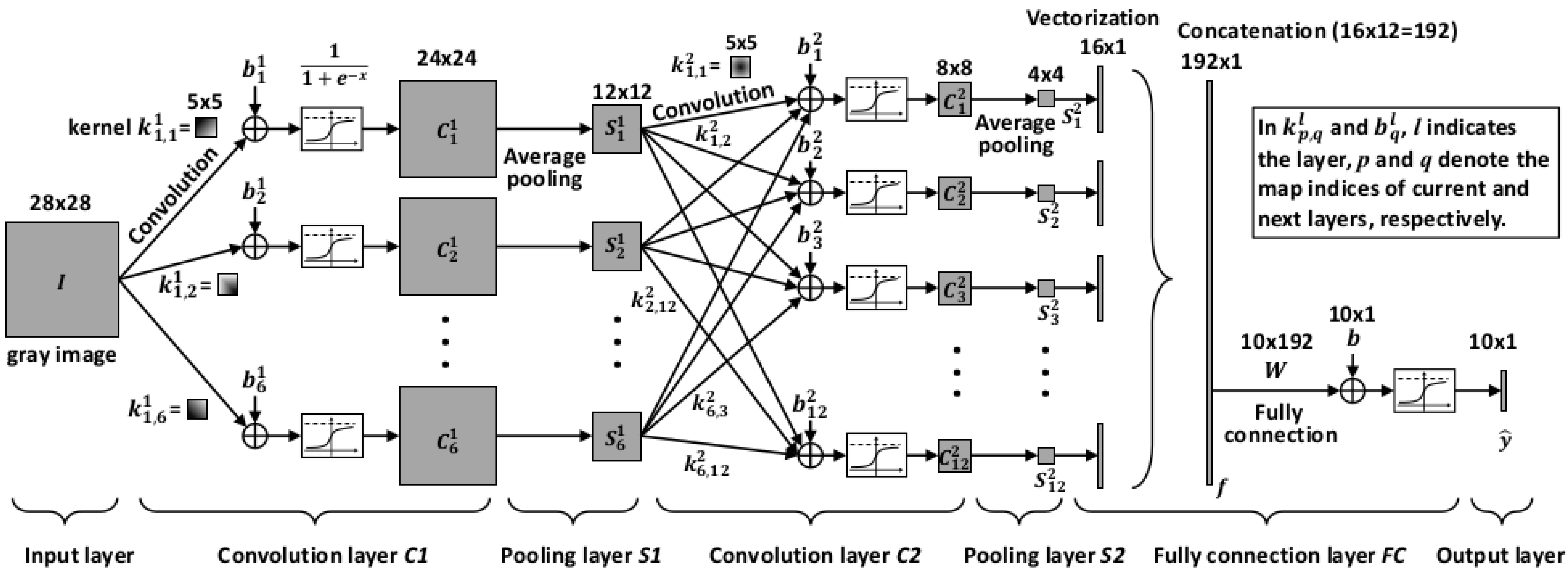


$\Delta k^1_{1,p}$ (**size** 5×5)

$$\Delta k^1_{1,p}(u,v) = \frac{\partial L}{\partial k^1_{1,p}(u,v)}$$

$$= \sum_{i=1}^{24}\sum_{j=1}^{24} \Delta C^1_p(i,j) \cdot C^1_p(i,j)\left(1 - C^1_p(i,j)\right) \cdot I(i-u, j-v)$$

CNN - Backpropagation



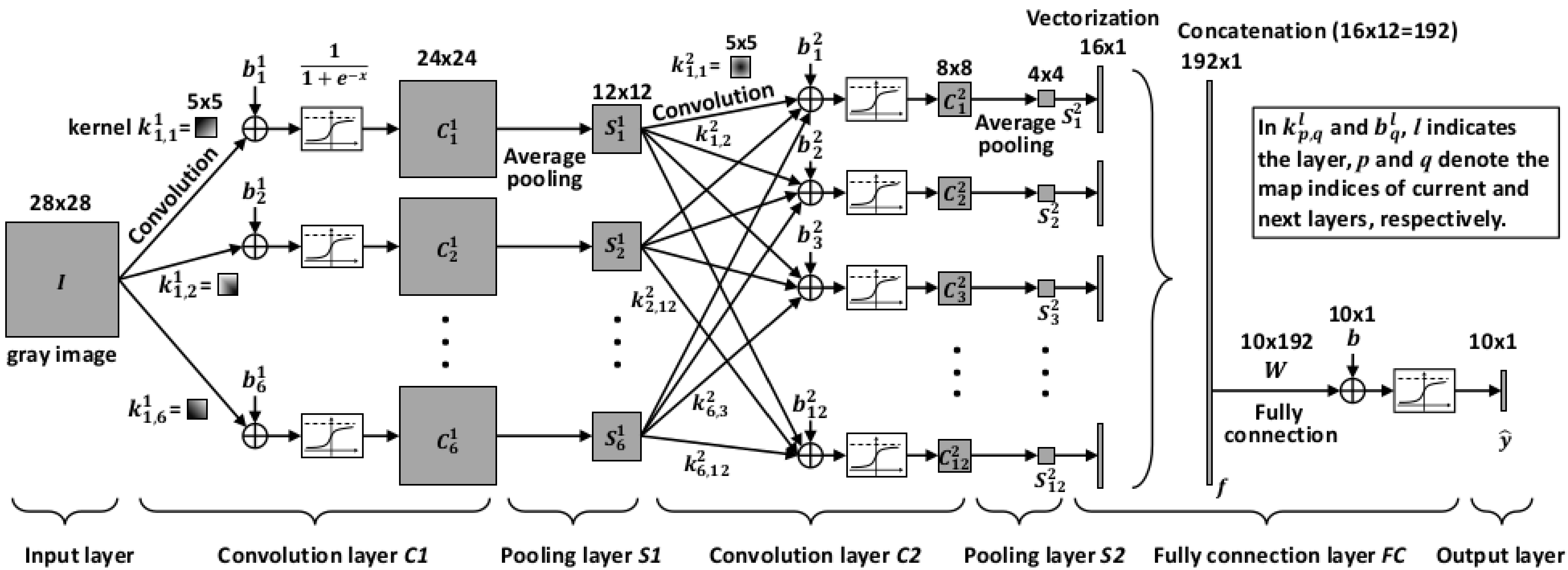
$\Delta k^1_{1,p}$ (size 5×5)

Rotacionando I de 180⁰

$$\Delta k^1_{1,p}(u,v) = \sum_{i=1}^{24} \sum_{j=1}^{24} I_{rot180}(u-i, v-j) \cdot \Delta C^1_{p,\sigma}(i,j)$$

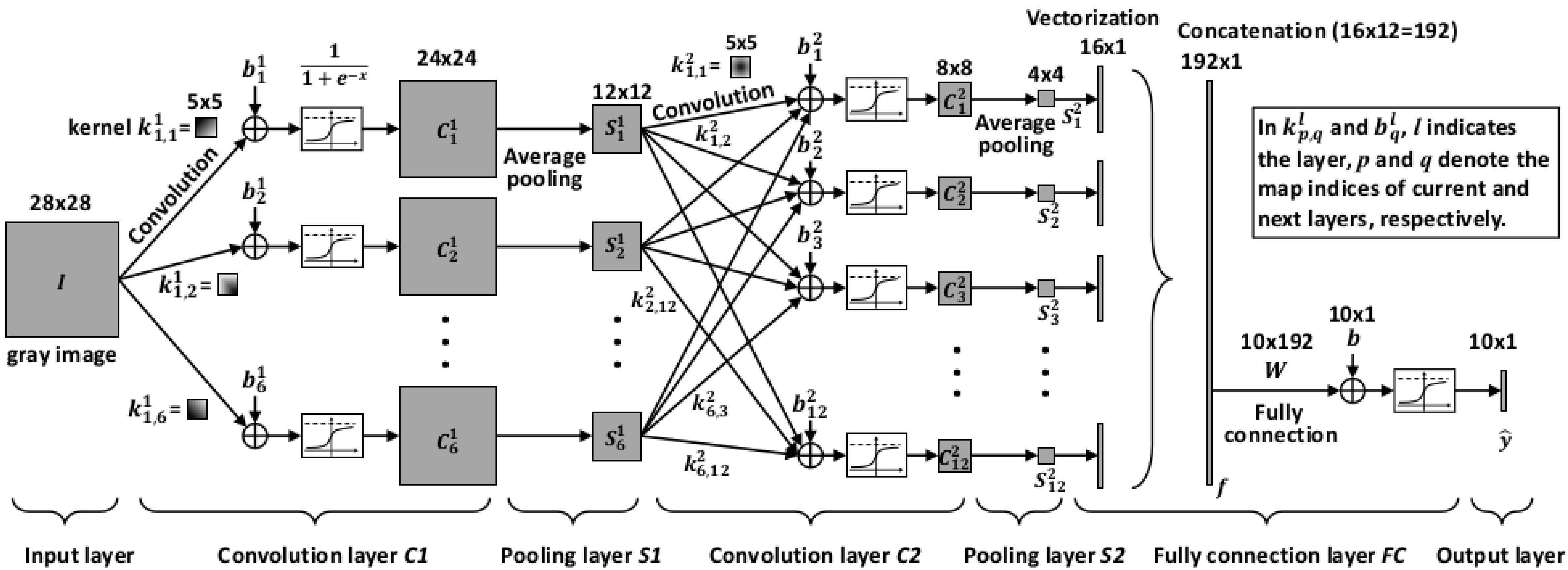
$$\Longrightarrow \Delta k^1_{1,p} = I_{rot180} * \Delta C^1_{p,\sigma}$$

CNN - Backpropagation



$$\Delta b_p^1 = \frac{\partial L}{\partial b_p^1} = \sum_{i=1}^{24}\sum_{j=1}^{24} \Delta C^1_{p,\sigma}(i,j)$$

CNN - Backpropagation



$$k^1_{1,p} \leftarrow k^1_{1,p} - \alpha \cdot \Delta k^1_{1,p}$$
$$b^1_p \leftarrow b^1_p - \alpha \cdot \Delta b^1_p$$
$$k^2_{p,q} \leftarrow k^2_{p,q} - \alpha \cdot \Delta k^2_{p,q}$$
$$b^2_q \leftarrow b^2_q - \alpha \cdot \Delta b^2_q$$
$$W \leftarrow W - \alpha \cdot \Delta W$$
$$b \leftarrow b - \alpha \cdot \Delta b$$

Referências

- https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5