Fundamentos de Aprendizaje Automático

Redes neuronales

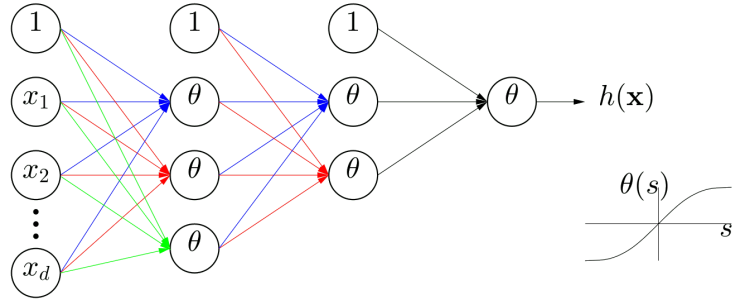
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería

UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Montevideo, 2021

1. Perceptrón multicapa
2. Aproximación versus generalización
3. Regularización y validación
4. Redes neuronales profundas
5. Redes neuronales convolucionales
6. Redes neuronales recurrentes

Perceptrón multicapa



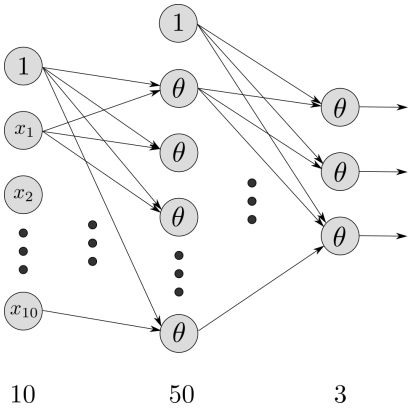
$$x_j^{(l)} = \theta(s_j^{(l)}) = \theta\left(\sum_{i=0}^{d^{(l-1)}} w_{ij}^{(l)} x_i^{(l-1)}\right)$$

$$x_1^{(0)} \dots x_{d^{(0)}}^{(0)} \to \cdots \to x_1^L = h(\mathbf{x})$$

$$w_{ij}^{(l)} = \begin{cases} 1 \leq l \leq L & \text{layers} \\ 0 \leq i \leq d^{(l-1)} L & \text{inputs} \\ 1 \leq j \leq d^{(l)} L & \text{outputs} \end{cases}$$

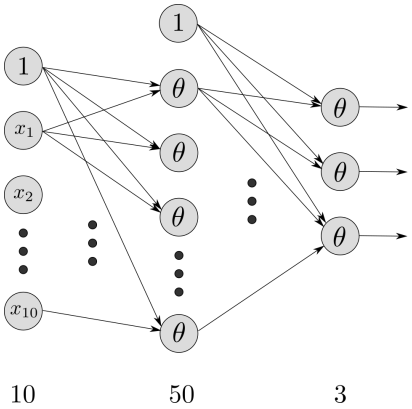
Ejercicio

Considere un perceptrón multicapa con 10 entradas, seguido de una capa oculta con 50 neuronas y una capa de salida con 3 neuronas. Sea m el tamaño del lote.



Ejercicio

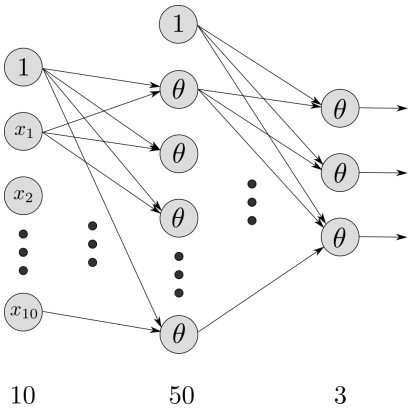
Considere un perceptrón multicapa con 10 entradas, seguido de una capa oculta con 50 neuronas y una capa de salida con 3 neuronas. Sea m el tamaño del lote.



1. ¿Cuál es la dimensión de la matriz $\mathbf{X}$?
2. ¿Cuál es la dimensión de $\mathbf{W}_h$ y $\mathbf{b}_h$?
3. ¿Cuál es la dimensión de $\mathbf{W}_o$ y $\mathbf{b}_o$?
4. ¿Cuál es la dimensión de la matriz $\mathbf{Y}$?
5. Escriba la ecuación para calcular la salida.

Ejercicio

Considere un perceptrón multicapa con 10 entradas, seguido de una capa oculta con 50 neuronas y una capa de salida con 3 neuronas. Sea m el tamaño del lote.



1. ¿Cuál es la dimensión de la matriz $\mathbf{X}$?
2. ¿Cuál es la dimensión de $\mathbf{W}_h$ y $\mathbf{b}_h$?
3. ¿Cuál es la dimensión de $\mathbf{W}_o$ y $\mathbf{b}_o$?
4. ¿Cuál es la dimensión de la matriz $\mathbf{Y}$?
5. Escriba la ecuación para calcular la salida.

$\mathbf{X} : (m \times 10)$, $\mathbf{W}_h : (10 \times 50)$, $\mathbf{b}_h : (50)$

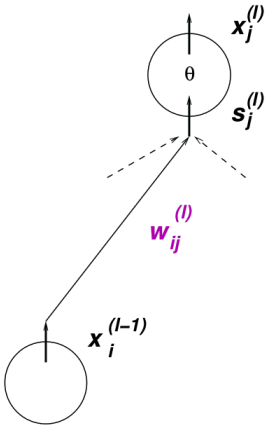
$\mathbf{W}_h : (50 \times 3)$, $\mathbf{b}_o : (3)$, $\mathbf{Y} : (m \times 3)$

$\mathbf{Y}^* = \theta\left(\theta(\mathbf{X}\mathbf{W}_h + \mathbf{b}_h)\mathbf{W}_o + \mathbf{b}_o\right)$

Propagación hacia atrás

Backpropagation*: gradiente descendente eficiente

$$\nabla e(\mathbf{w}) : \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} \; \forall i, j, l \qquad \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} = \frac{\partial e(\mathbf{w})}{\partial s_j^{(l)}} \times \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}}$$



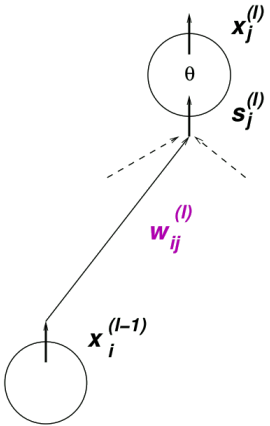
*D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986

Propagación hacia atrás

Backpropagation*: gradiente descendente eficiente

$$\nabla e(\mathbf{w}) : \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} \; \forall i, j, l \qquad \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} = \frac{\partial e(\mathbf{w})}{\partial s_j^{(l)}} \times \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}}$$

$$\frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} = x_i^{(l-1)} \qquad \frac{\partial e(\mathbf{w})}{\partial s_j^{(l)}} = \delta_j^{(l)}$$



*D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986

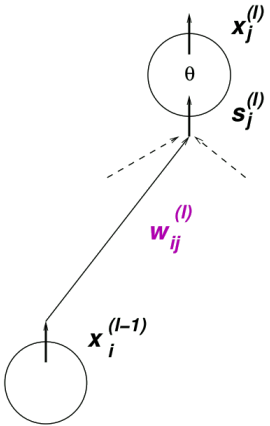
Propagación hacia atrás

Backpropagation*: gradiente descendente eficiente

$$\nabla e(\mathbf{w}) : \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} \; \forall i, j, l \qquad \frac{\partial e(\mathbf{w})}{\partial w_{ij}^{(l)}} = \frac{\partial e(\mathbf{w})}{\partial s_j^{(l)}} \times \frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}}$$

$$\frac{\partial s_j^{(l)}}{\partial w_{ij}^{(l)}} = x_i^{(l-1)} \qquad \frac{\partial e(\mathbf{w})}{\partial s_j^{(l)}} = \delta_j^{(l)}$$

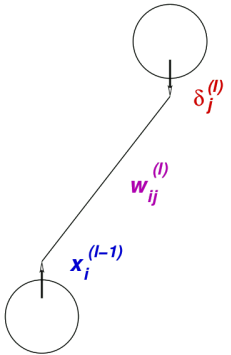
se calcula $\delta_1^{(L)}$ (última capa) y se propaga hacia atrás



*D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986

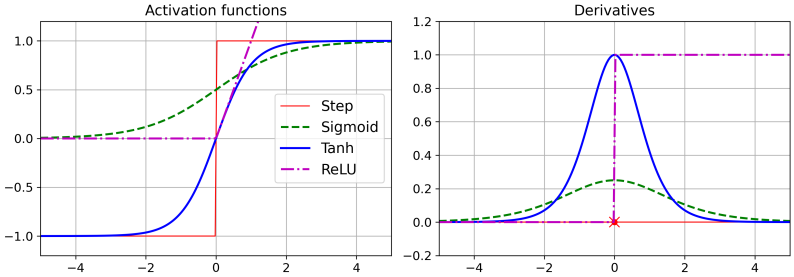
Propagación hacia atrás

1: Initialize all weights $w_{ij}^{(l)}$ at random
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Pick $n \in \{1, 2, \cdots, N\}$
4: *Forward:* Compute all $x_j^{(l)}$
5: *Backward:* Compute all $\delta_j^{(l)}$
6: Update the weights: $w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta\, x_i^{(l-1)} \delta_j^{(l)}$
7: Iterate to the next step until it is time to stop
8: Return the final weights $w_{ij}^{(l)}$



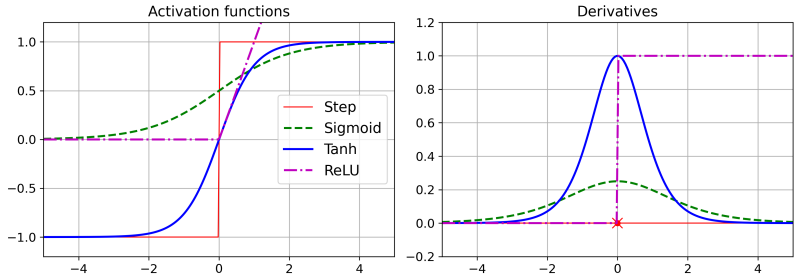
Funciones de activación

- Sigmoid : $\theta(x) = \frac{1}{1+e^{-s}}$
- Tanh : $\theta(x) = \tanh(x) = \frac{e^{s}-e^{-s}}{e^{s}+e^{-s}}$
- ReLU : $\theta(x) = \max(0, x)$



Funciones de activación

- Sigmoid : $\theta(x) = \frac{1}{1+e^{-s}}$
- Tanh : $\theta(x) = \tanh(x) = \frac{e^{s}-e^{-s}}{e^{s}+e^{-s}}$
- ReLU : $\theta(x) = \max(0, x)$



Pregunta

¿Para qué necesitamos funciones de activación? ¿Sería suficiente con funciones lineales?

Aproximación versus generalización

Teorema de aproximación universal

Una red neuronal prealimentada (feed-forward) con una única capa oculta y un número finito de neuronas, puede aproximar cualquier función continua en un espacio compacto de $\mathbb{R}^n$.*†

*G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989

†K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991

‡Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in NIPS*, vol. 30, 2017

Aproximación versus generalización

Teorema de aproximación universal

Una red neuronal prealimentada (feed-forward) con una única capa oculta y un número finito de neuronas, puede aproximar cualquier función continua en un espacio compacto de $\mathbb{R}^n$.*†

- el ancho de la red debe ser exponencialmente grande
- aproximación universal para redes profundas de ancho acotado ‡

*G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989

†K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991

‡Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in NIPS*, vol. 30, 2017

Aproximación versus generalización

Teorema de aproximación universal

Una red neuronal prealimentada (feed-forward) con una única capa oculta y un número finito de neuronas, puede aproximar cualquier función continua en un espacio compacto de $\mathbb{R}^n$.*†

- el ancho de la red debe ser exponencialmente grande
- aproximación universal para redes profundas de ancho acotado ‡

Generalización

Las redes neuronales tienen tendencia al sobreajuste.

*G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989

†K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991

‡Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in NIPS*, vol. 30, 2017

Aproximación versus generalización

Teorema de aproximación universal

Una red neuronal prealimentada (feed-forward) con una única capa oculta y un número finito de neuronas, puede aproximar cualquier función continua en un espacio compacto de $\mathbb{R}^n$.*†

- el ancho de la red debe ser exponencialmente grande
- aproximación universal para redes profundas de ancho acotado ‡

Generalización

Las redes neuronales tienen tendencia al sobreajuste.

- resultados teóricos sobre error de generalización (d_{VC})
- regularización y validación para prevenir sobreajuste

*G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989

†K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991

‡Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in NIPS*, vol. 30, 2017

Regularización y validación

- Ridge (weight decay):

  Error con penalización sobre la complejidad del modelo.

$$E_{aug}(\mathrm{w}) = E_{in}(\mathrm{w}) + \frac{\lambda}{N} \sum_{i,j,l} (w_{ij}^{(l)})^2$$

Regularización y validación

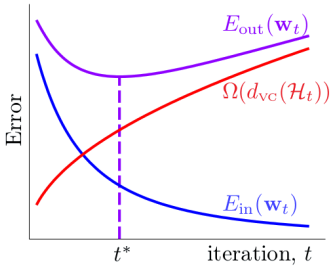
- Ridge (weight decay):

  Error con penalización sobre la complejidad del modelo.

$$E_{aug}(\mathrm{w}) = E_{in}(\mathrm{w}) + \frac{\lambda}{N}\sum_{i,j,l}(w_{ij}^{(l)})^2$$

- Early stopping:

  Se detiene el entrenamiento antes de sobreajustar, evaluando en validación.



Regularización y validación

- Dropout

  Ignorar aleatoriamente algunas neuronas (10 a 50%) en cada paso de entrenamiento. Se obtiene una red menos sensible a cambios en la entrada y menos sobreajustada.

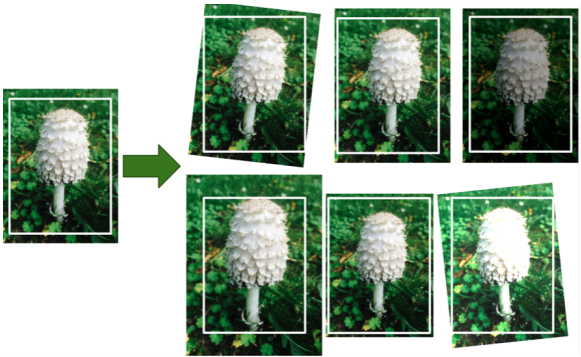
Regularización y validación

- **Dropout**

  Ignorar aleatoriamente algunas neuronas (10 a 50%) en cada paso de entrenamiento. Se obtiene una red menos sensible a cambios en la entrada y menos sobreajustada.

- **Data augmentation**

  Aumentar artificialmente los datos de entrenamiento mediante transformaciones (realistas).

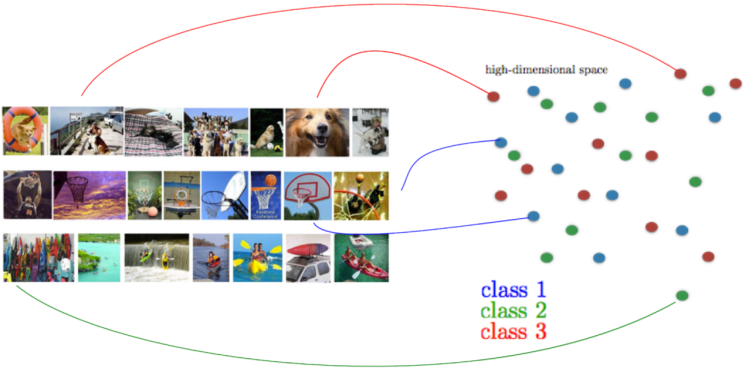


Redes neuronales profundas: aprendizaje profundo

Una clase de representaciones paramétricas no lineales capaces de codificar características (o conocimiento) del problema y de ser optimizadas de forma eficiente (a enorme escala) usando métodos de descenso por gradiente estocástico.

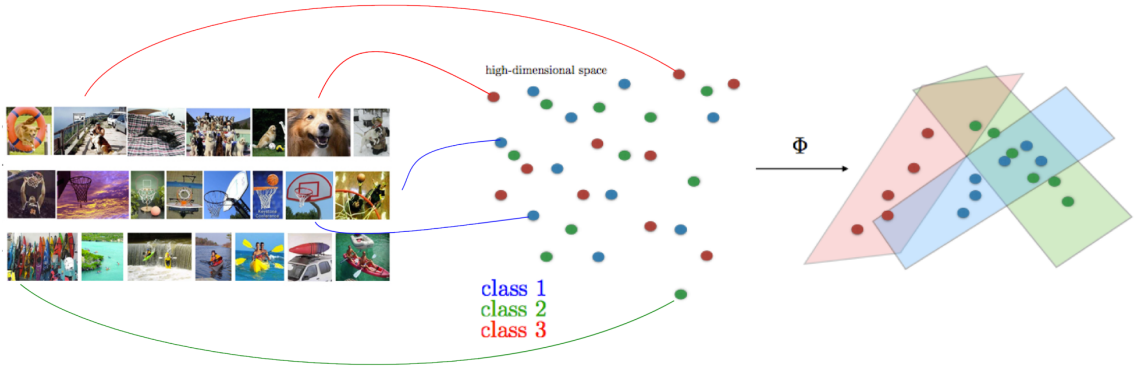
Redes neuronales profundas: aprendizaje profundo

Una clase de representaciones paramétricas no lineales capaces de codificar características (o conocimiento) del problema y de ser optimizadas de forma eficiente (a enorme escala) usando métodos de descenso por gradiente estocástico.



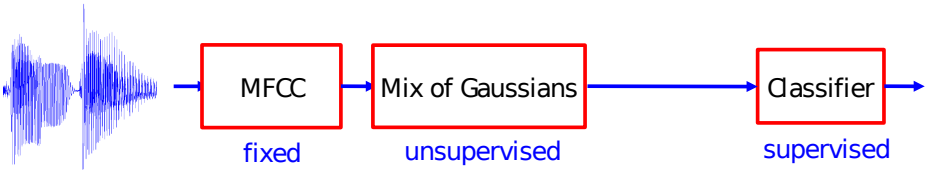
Redes neuronales profundas: aprendizaje profundo

Una clase de representaciones paramétricas no lineales capaces de codificar características (o conocimiento) del problema y de ser optimizadas de forma eficiente (a enorme escala) usando métodos de descenso por gradiente estocástico.

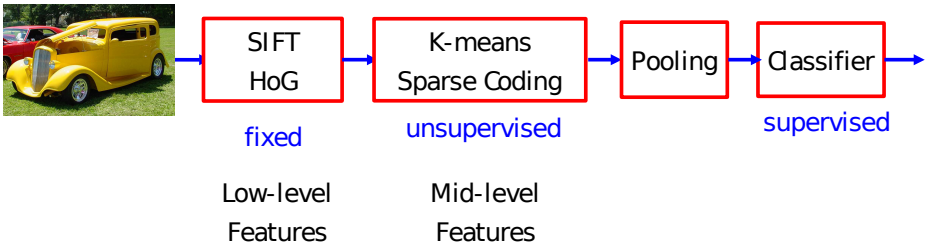


Arquitecturas de reconocimiento de patrones

Reconocimiento de audio, principios de los 90 hasta el 2011:

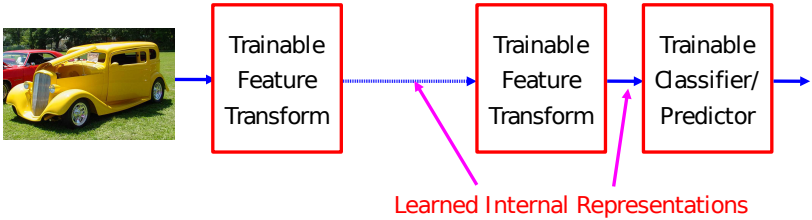


Reconocimiento de imágenes, principios del 2000 hasta el 2011:



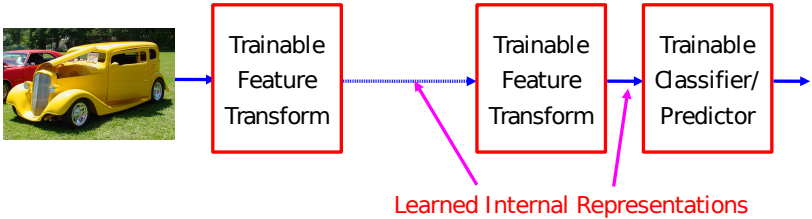
Ejemplo tomado de charlas de Y. Lecun.

Representaciones jerárquicas adaptivas



- Una jerarquía de representaciones (capas) con mayores niveles de abstracción
- Cada capa es una representación de características adaptiva (entrenable)

Representaciones jerárquicas adaptivas

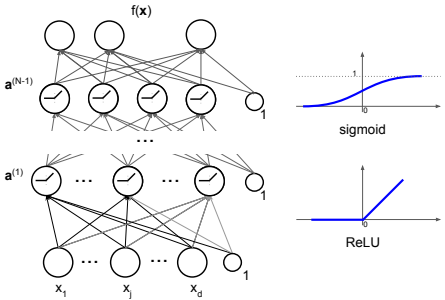


- Una jerarquía de representaciones (capas) con mayores niveles de abstracción
- Cada capa es una representación de características adaptiva (entrenable)

- Imágenes:
  - píxeles → bordes → partes → objetos

Redes neuronales profundas: funciones de activación

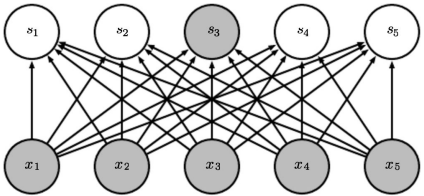
- el gradiente puede desvanecerse o explotar en el entrenamiento
- inicialización inadecuada y función de activación que satura



- la función ReLU: $h(x) = \max(0, x)$ da mejores resultados
- Batch Normalization: escalado y corrimiento de las entradas de cada capa (con parámetros aprendidos).

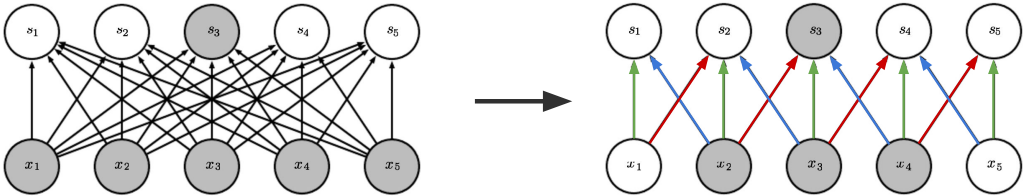
Redes neuronales convolucionales (CNN)

- Para datos en altas dimensiones, las redes neuronales *totalmente conectadas*, tal como vimos hasta ahora requieren un gran número de parámetros
- Por ejemplo, procesar imágenes de Imagenet (tamaño $224 \times 224 \times 3$) para clasificar entre 1000 clases, con una sola capa requeriría 150 millones de parámetros.

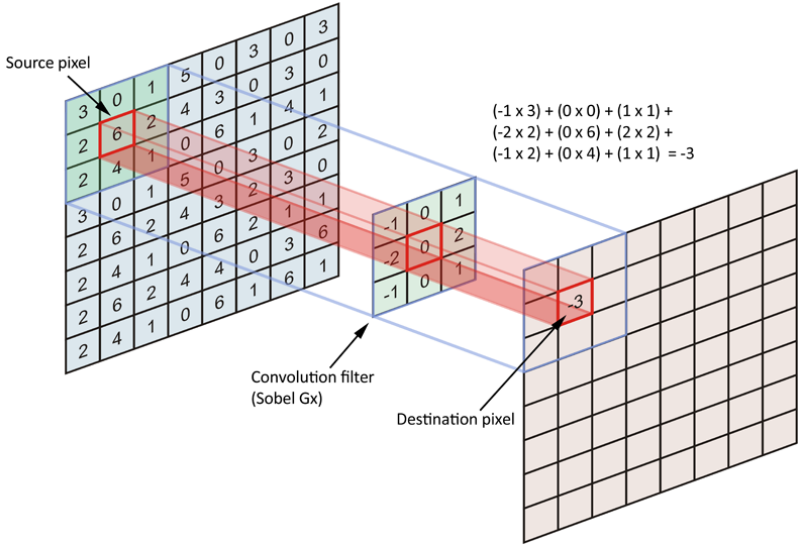


Redes neuronales convolucionales (CNN)

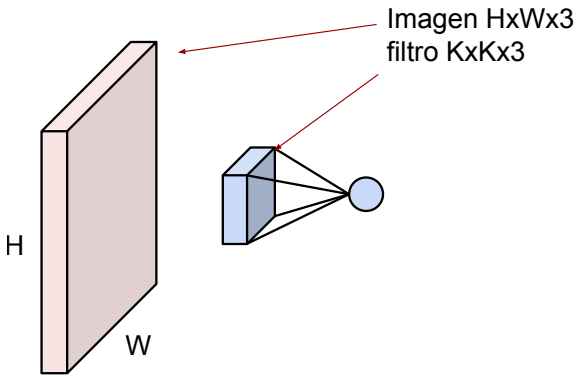
- Para datos en altas dimensiones, las redes neuronales *totalmente conectadas*, tal como vimos hasta ahora requieren un gran número de parámetros
- Por ejemplo, procesar imágenes de Imagenet (tamaño $224 \times 224 \times 3$) para clasificar entre 1000 clases, con una sola capa requeriría 150 millones de parámetros.
- Las redes de convolución surgen para evitar este problema haciendo cálculos más eficientes y explotando invarianzas inherentes a las imágenes



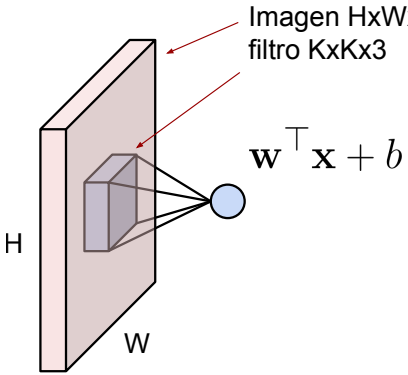
Convolución



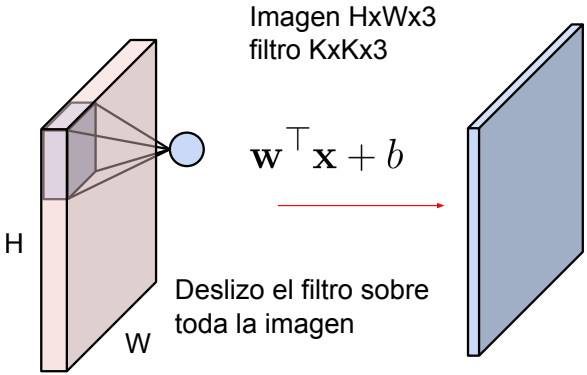
Capa de convolución



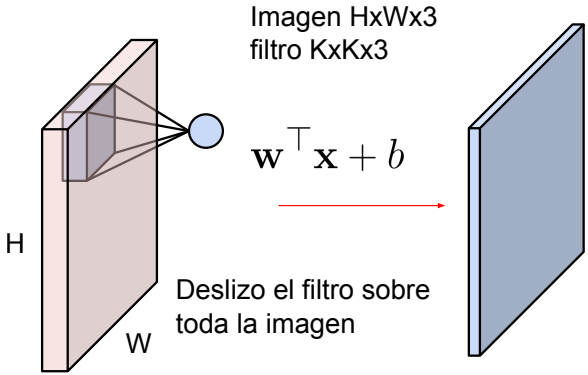
Capa de convolución



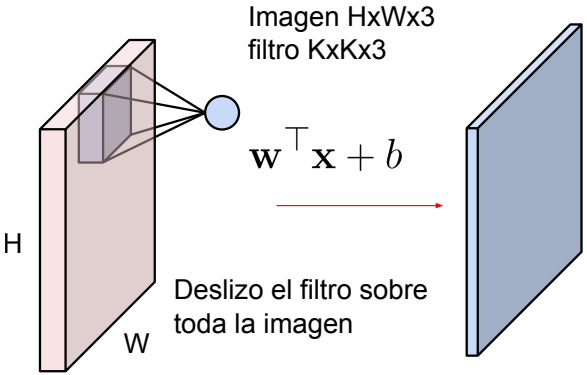
Capa de convolución



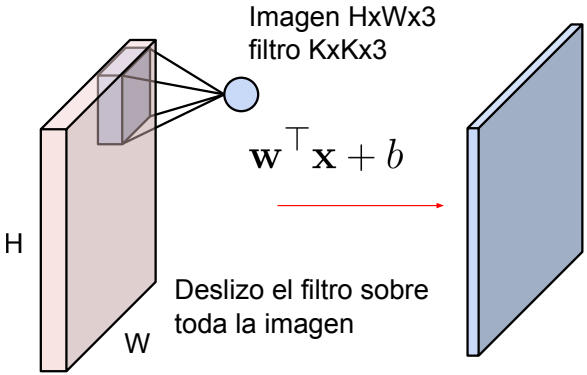
Capa de convolución



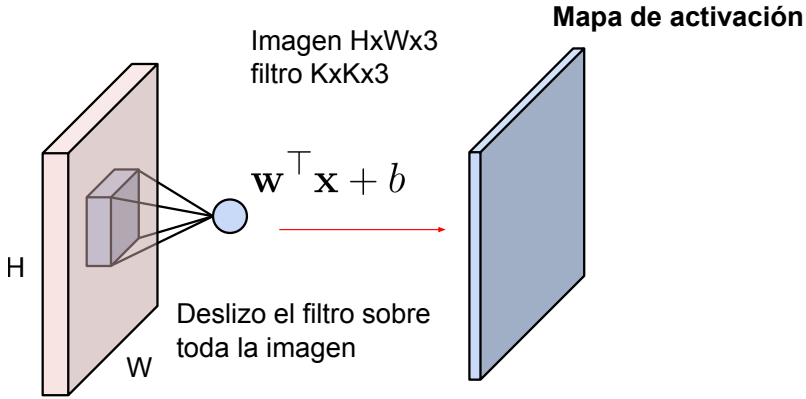
Capa de convolución



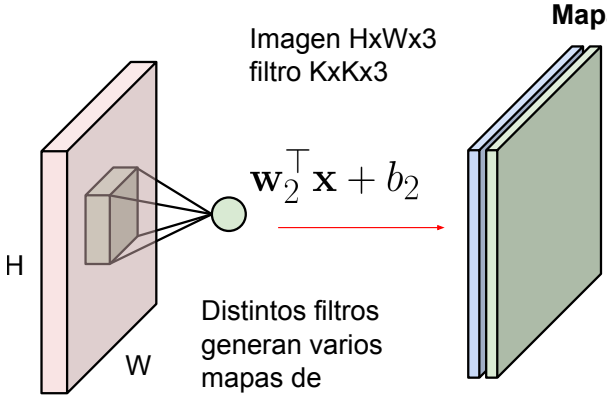
Capa de convolución



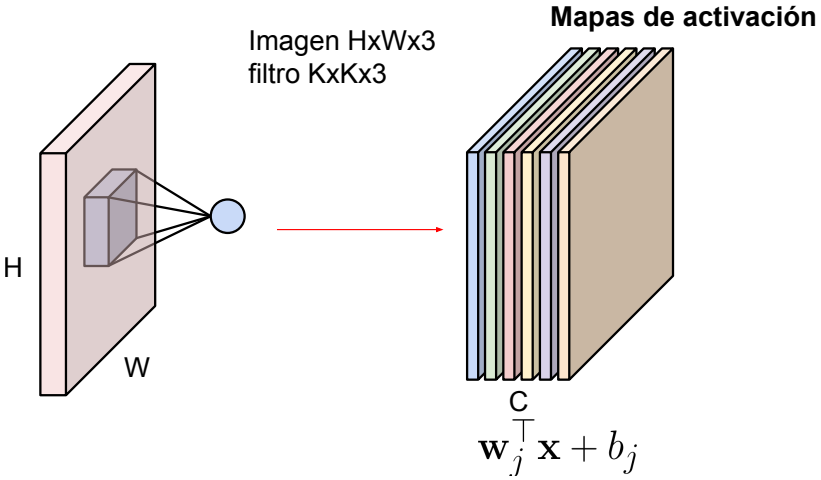
Capa de convolución



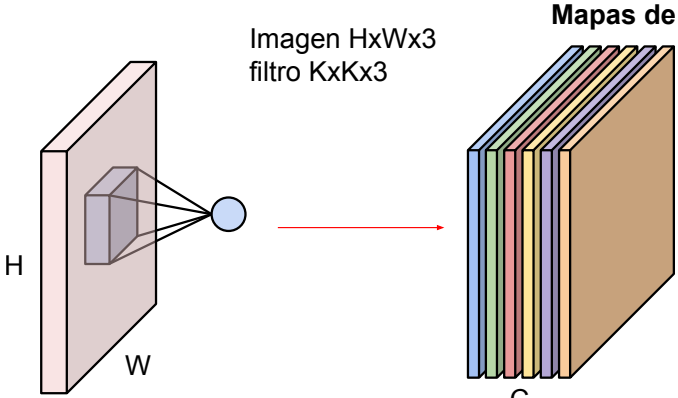
Capa de convolución



Capa de convolución



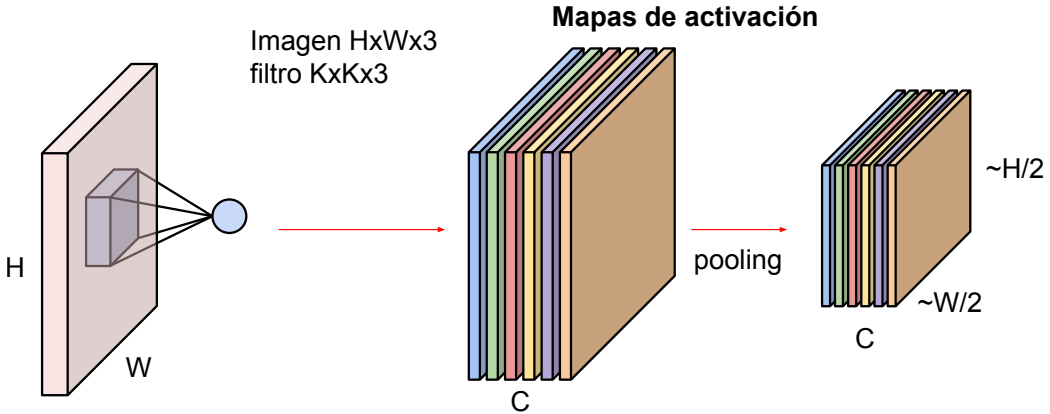
Capa de convolución



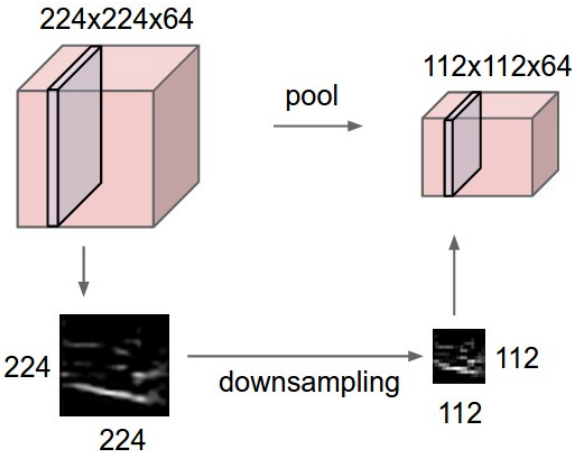
Aplico la función de activación (e.g. ReLU)

$$g(\mathbf{w}_j^\top \mathbf{x} + b_j)$$

Capa de convolución



Pooling



Arquitectura típica de una CNN

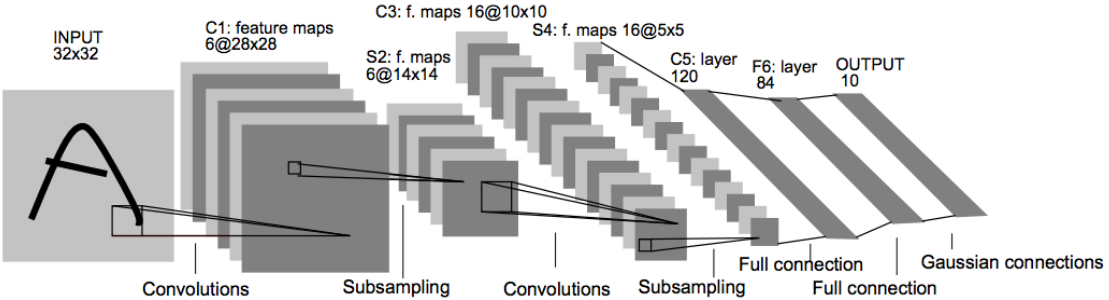
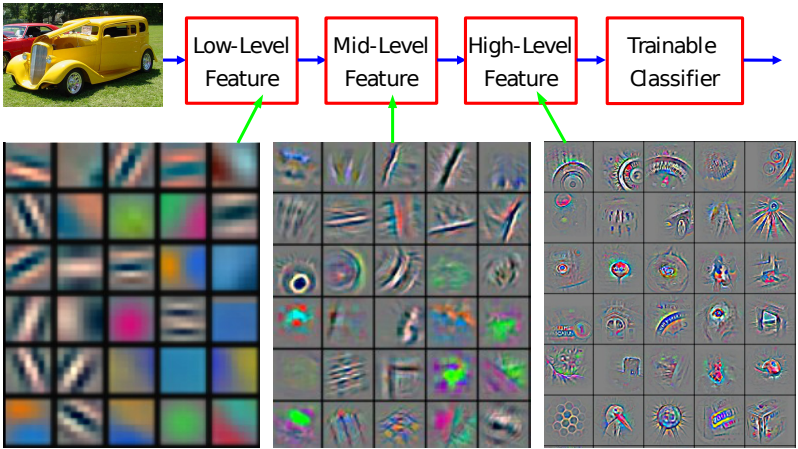


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- Arquitectura LeNet-5, propuesta por *Yann Lecun et al. 1998*

¿Qué filtros aprende una CNN?



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Slide de Y. Lecun

¿Qué filtros aprende una CNN?

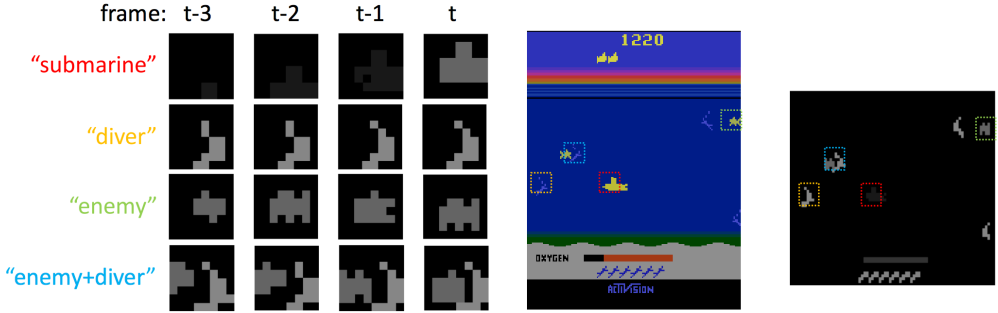


Figure 2: Visualization of the first-layer features learned from Seaquest. (Left) visualization of four first-layer filters; each filter covers four frames, showing the spatio-temporal template. (Middle) a captured screen. (Right) gray-scale version of the input screen which is fed into the CNN. Four filters were color-coded and visualized as dotted bounding boxes at the locations where they get activated. This figure is best viewed in color.

Guo et al. 2014

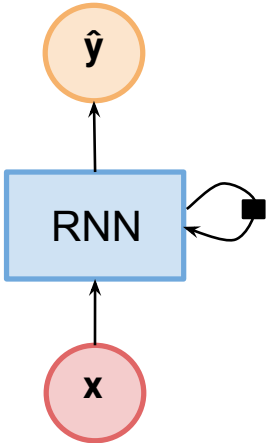
Redes neuronales recurrentes (RNN)

- para problemas secuenciales donde
  - largo de las secuencias de entrada/salida variable
  - quiero guardar información de la *historia*
- guardan un estado oculto h_t que depende de la entrada actual y el estado anterior h_{t-1}

$$h_t = f_w(h_{t-1}, x_t)$$

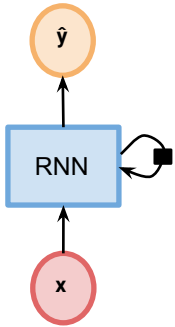
- ejemplo básico de RNN

$$\begin{aligned} h_t &= \tanh\left(W_{hh} h_{t-1} + W_{hx} x_t\right) \\ \hat{y}_t &= W_{hy} h_t \end{aligned}$$



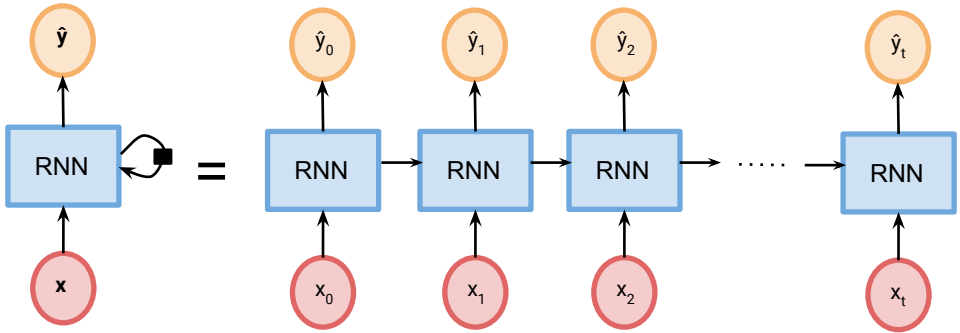
Redes neuronales recurrentes (RNN)

- Una red neuronal recurrente puede pensarse como múltiples copias de la misma red donde cada una le pasa un mensaje (estado) al sucesor



Redes neuronales recurrentes (RNN)

- Una red neuronal recurrente puede pensarse como múltiples copias de la misma red donde cada una le pasa un mensaje (estado) al sucesor

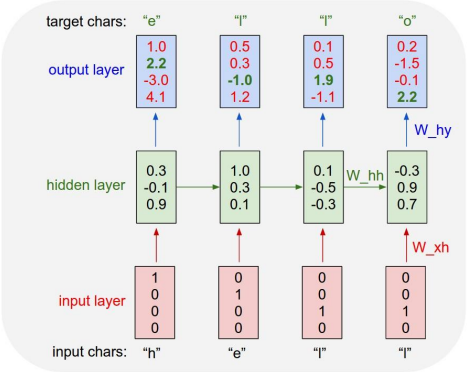


Modelado de lenguaje a nivel de carácter

Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”



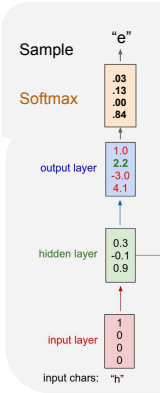
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Modelado de lenguaje a nivel de carácter

Example:
Character-level
Language Model
Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



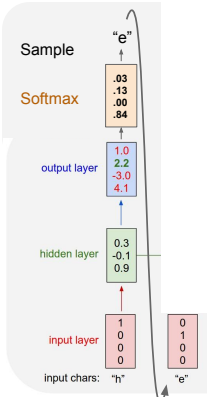
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Modelado de lenguaje a nivel de carácter

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



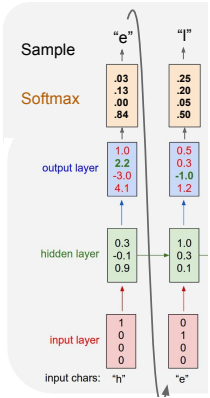
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Modelado de lenguaje a nivel de carácter

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



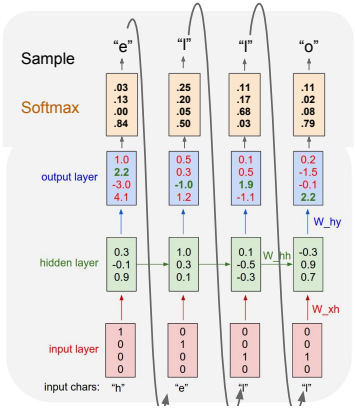
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Modelado de lenguaje a nivel de carácter

Example:
Character-level
Language Model
Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

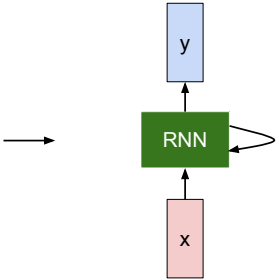
Modelado de lenguaje a nivel de carácter

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.



Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Modelado de lenguaje a nivel de carácter

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

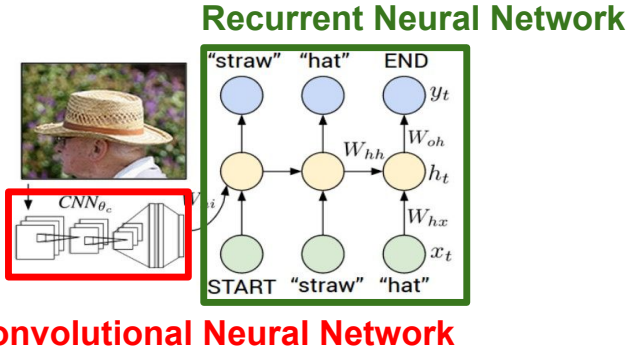
```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



Andrej Karpathy Li Fei-Fei
"Deep Visual-Semantic Alignments for Generating Image Descriptions"

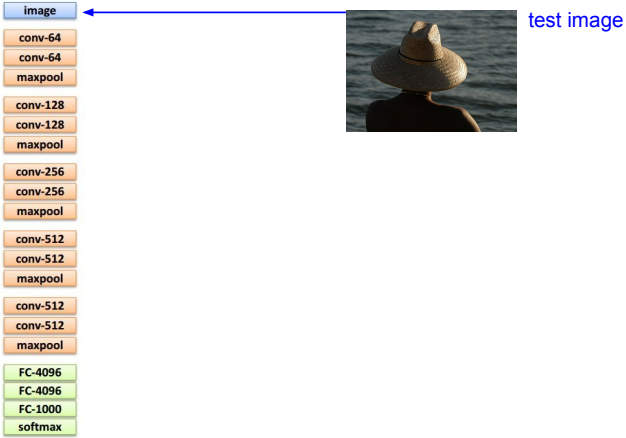
Generación automática de leyenda en imágenes

test image

This image is CC0 public domain

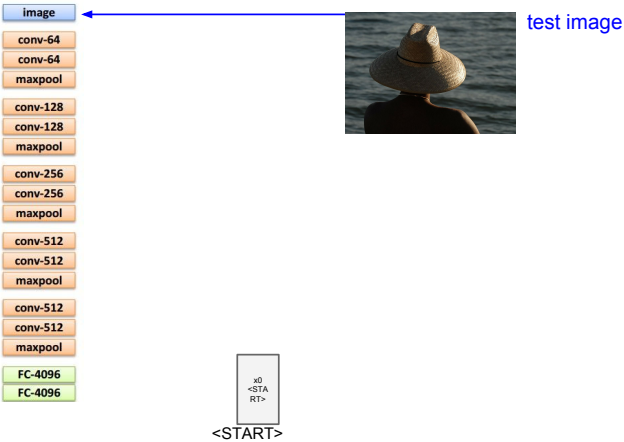
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



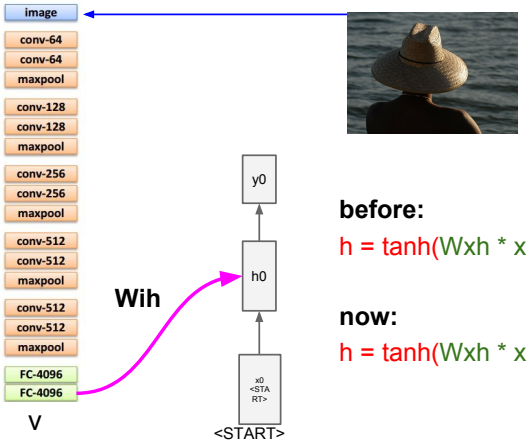
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



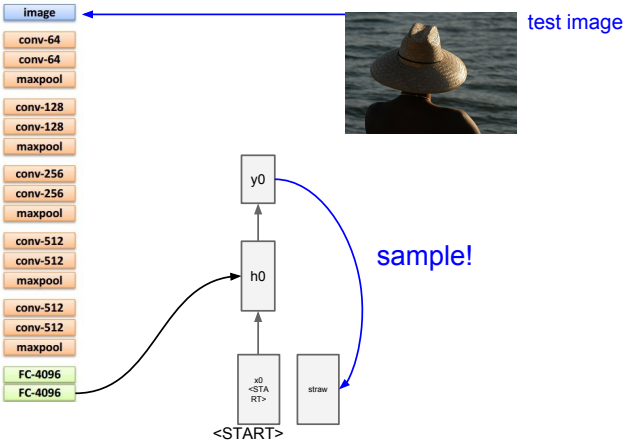
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



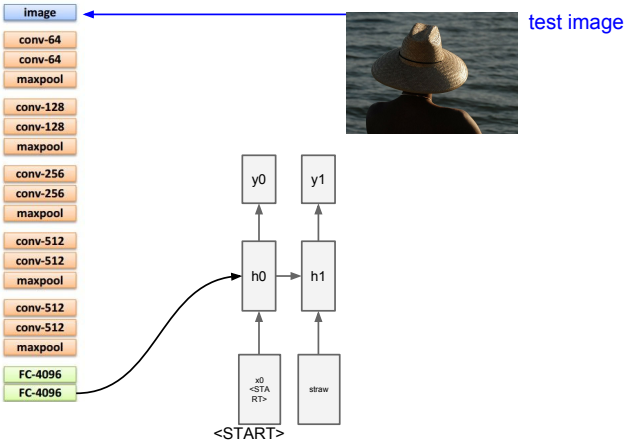
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



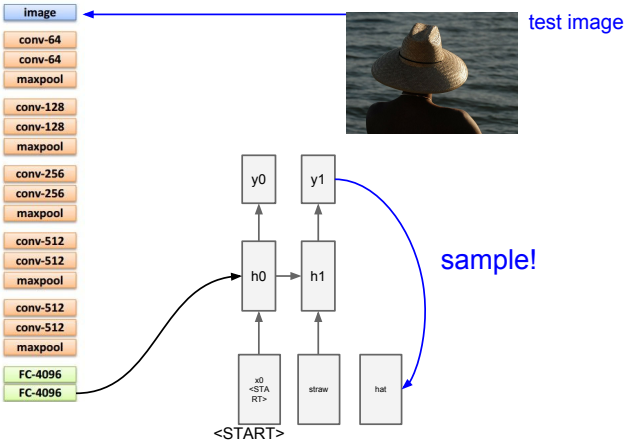
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



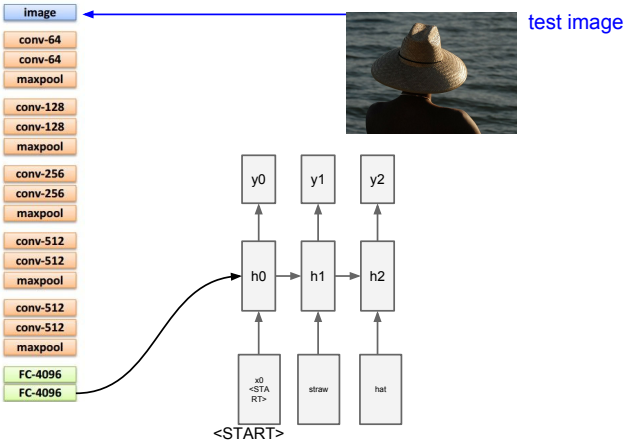
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



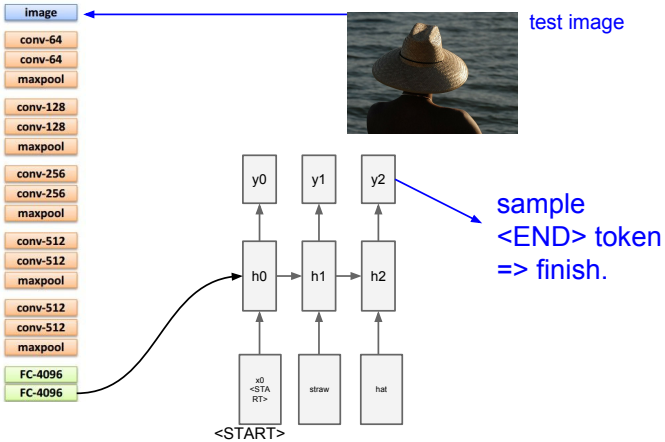
Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes



Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes

Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Generación automática de leyenda en imágenes

Image Captioning: Failure Cases

Captions generated using neuraltalk2
All images are CC0 Public domain: fur coat, handstand, spider web, baseball

A woman is holding a cat in her hand

A person holding a computer mouse on a desk

A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball

Slides tomadas de cs231n (Stanford) - Fei-Fei Li & Justin Johnson & Serena Yeung

Referencias


D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.


G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.


K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.


Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Advances in NIPS*, vol. 30, 2017.


A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. O'Reilly Media, Inc., 2019.