

SCC0270/SCC5809 - Redes Neurais

Aula - Deep Learning

Profa. Dra. Roseli Aparecida Francelin Romero
SCC - ICMC - USP

2018

Sumário

1 Introdução

- Motivação
- Deep Learning
- Contextualização histórica

2 Convolutional Neural Networks (CNN)

3 Funções Utilizadas

Sumário

- 1 **Introdução**
 - Motivação
 - Deep Learning
 - Contextualização histórica

- 2 Convolutional Neural Networks (CNN)
- 3 Funções Utilizadas

Introdução

- Tarefas formais e abstratas: difíceis para humano, porém fáceis para computadores.
 - Com pouco conhecimento de mundo, o *Deep Blue* venceu um campeão de xadrez.
- Desafios da inteligência artificial: problemas fáceis para humanos resolverem, porém difíceis de descrever formalmente (i.e. problemas **intuitivos**).
- Conhecimento *hard-coded* é limitado e demanda muito esforço e tempo para ser implementado.
 - Por isso, o computador precisa adquirir seu próprio conhecimento, extraíndo características de dados brutos (aprendizado de máquina - AM).

Introdução

- Desempenho do algoritmo de AM depende da **representação** do conjunto de características.
 - **Exemplo:** humanos calculam soma muito mais rapidamente com algarismos indo-arábicos do que com algarismos romanos.
- Muitos problemas de inteligência artificial são resolvidos descobrindo-se características corretas e aplicando-se um algoritmo simples de aprendizado de máquina.
- Determinar as características nem sempre é trivial.
 - Muitos pesquisadores dedicaram e dedicam suas carreiras à extração de características em imagens, áudio, etc.

Introdução

- **Exemplo:** *carro* tem *rodas*, mas como reconhecer uma roda em termos de pixels em uma imagem?
- **Representation Learning:** aprender representações automaticamente, em vez de defini-las manualmente.
 - *Autoencoders* codificam uma entrada e depois a decodificam, obtendo uma nova representação.
 - Em alguns casos, essa técnica consiste em detectar **fatores de variância**: atributos "linearmente independentes" que descrevem a informação analisada.
- Muitos fatores de variância somente podem ser obtidos com uma compreensão sofisticadíssima, quase humana, dos dados (e.g. sotaque).
 - Nesses casos, *representation learning* não ajuda.

Sumário

1 Introdução

- Motivação
- **Deep Learning**
- Contextualização histórica

2 Convolutional Neural Networks (CNN)

3 Funções Utilizadas

Introdução

Deep Learning

Aprendizado composto por uma **hierarquia de conceitos**, em que cada conceito é definido em relação a conceitos mais simples.

Introdução

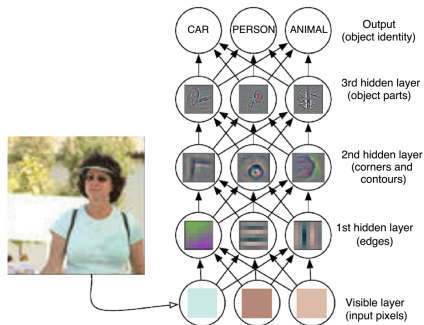


Figura 1: Ilustração de um modelo de *deep learning*.

Introdução

- Deep Neural Networks (DNNs) é a área mais ativa de pesquisas de AM nos dias de hoje.
- O sucesso se deve a excelente performance obtida para classificação de dígitos manuscritos e detecção de faces.
- O sucesso das DNNs tem sido graças ao uso das GPUs, que tem se tornado a plataforma de escolha para treinamento.
- Empresas Comerciais têm investido agressivamente nesta tecnologia (DNNs + GPU).
- DNNs tipicamente tem muitas cam. que não são totalmente conectadas.

Introdução

- Há duas formas de determinar a **profundidade** de um modelo:
 - 1 Número de instruções sequenciais executadas (maior caminho em um fluxograma). O número de passos pode diferir de acordo com a linguagem ou com as funções disponíveis, conforme ilustra a Figura 2.
 - 2 Número de relações entre conceitos (e.g. número de camadas em uma MLP).

Introdução

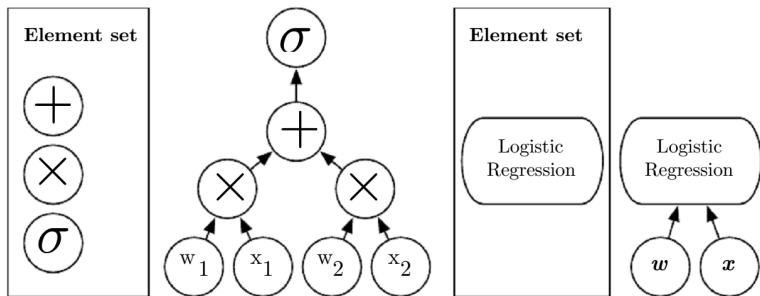


Figura 2: Grafos representando um mapeamento, em que cada nó é responsável por uma operação. A profundidade varia se permitirmos, como operadores, somente as operações de adição, subtração e função sigmoide (esquerda, profundidade 3) ou se provermos um elemento que, sozinho, realiza *regressão linear* (direita, profundidade 1).

Introdução

- Além de não haver uma abordagem correta para utilizar, as pessoas podem definir cada uma delas de maneiras diferentes.
 - **Não há** um único valor correto para profundidade.
- Diz-se que o aprendizado é profundo quando sua arquitetura é mais profunda em relação às outras técnicas existentes.

Introdução

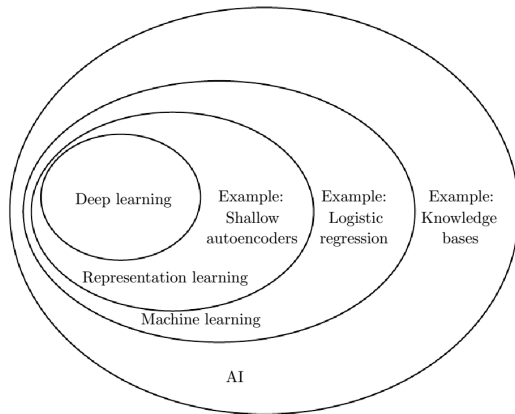
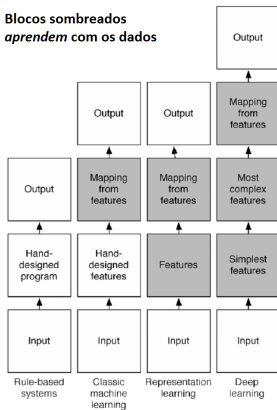


Figura 3: Diagrama de Venn com a relação entre inteligência artificial, aprendizado de máquina, *representation learning* e *deep learning*.

Introdução

**Blocos sombreados
aprendem com os dados**



Sumário

- 1 **Introdução**
 - Motivação
 - Deep Learning
 - Contextualização histórica

- 2 Convolutional Neural Networks (CNN)
- 3 Funções Utilizadas

Contextualização histórica

- Três ondas históricas de pesquisa em redes neurais artificiais:
- **Cibernética:** *Perceptron*, *Adaline*, neurônio de McCulloch-Pitts. Enfraqueceu devido a suas limitações Minsky and Papert [1969].
- **Conexionismo:** representação distribuída, MLP, *backpropagation*. Enfraqueceu (não tanto) porque não cumpriu promessas ambiciosas e porque máquinas de *kernel*, surgidas em seguida, apresentaram garantias teóricas.
- **Deep Learning:** treinar máquinas de *kernel* implica tempo e memória quadráticos com o tamanho da base de dados. As bases de dados cresceram, eliminando as vantagens dessa abordagem.
 - Redes neurais profundas (DNNs) são adequadas para esses problemas.

The graph illustrates the cyclical nature of AI research over time. The x-axis represents years from 1940 to 2020. The y-axis represents the level of research activity. Three peaks are labeled: 'cybernetics' around 1960, 'connectionism' around 1990, and 'deep learning' around 2015. The graph shows a general upward trend with periodic dips.

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

Contextualização histórica

- As primeiras redes neurais profundas a obterem ampla repercussão foram as redes baseadas em pré-treinamento não supervisionado (ano de 2006) [Hinton and Salakhutdinov, 2006].
 - *Deep Belief Networks* (DBN).
 - *Stacked autoencoders*.
- Entretanto, foi o redescobrimiento das redes neurais convolucionais (CNN) [LeCun et al., 1998] que tornaram esse tema um dos principais tópicos em aprendizado de máquina.
 - Tecnologias de processamento paralelo em GPUs e grandes bases de dados permitiram que essas redes fossem usadas plenamente.

Base de dados ImageNet

- Apresentada pela primeira vez em 2009, na *Conference on Computer Vision and Pattern Recognition* (CVPR).
- Imagens descritas por *synsets*, i.e., conceitos descritivos da base hierárquica *WordNet*.
- 14.197.122 imagens.
- 21.841 *synsets* não vazios.
 - Podem ser reduzidos em conceitos menos específicos (e.g., de "pardal" para "pássaro"), diminuindo o número total de classes.
- 1.034.908 imagens com anotações em *bounding box*.

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- Desafio anual para classificação da base de dados ImageNet, iniciado em 2010.
- Em 2012, pela primeira vez, uma CNN foi usada para esse propósito (AlexNet).
- Erro de teste *top 5* de 15,4%.
 - O segundo colocado obteve 26,2%.
- Esse resultado chocou a comunidade de visão computacional, tornando as redes neurais profundas o padrão em classificação de imagens e dando origem à onda atual de desenvolvimento nessa área.

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN)

- Introdução

- Inspiração biológica
- Características da rede
- Exemplo ilustrativo
- Rede LeNet

3 Funções Utilizadas

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN) - Introdução

- Inspiração biológica
- Características da rede
- Exemplo ilustrativo
- Rede LeNet

3 Funções Utilizadas

Características

- Camada Convolutacional: características relevantes
- Sub-amostragem (Sub-sampling): proporciona certa invariância em relação a posição do filtro.

Aplicações

- tarefas de detecção, segmentação e reconhecimento de objetos.
- reconhecimento de faces [Taigman et. all] e [Li et all.], expressões faciais [Yu and Zhang], detecção de placas de trânsito[Ciresan et all.], de texto e de pedestres [Sermanet et all.], de corpos humanos [Tompson et all.] em imagens naturais.

O sucesso é devido aos Avanços

- 1 : hardwares de videos (GPUs): treinamento de grandes redes neurais profundas em tempo aceitável.
- 2 : Função de Ativação - **ReLu** (rectified linear unit);
aprendizado mais rápido com redes com muitas camadas, em vez de funções mais suaves como sigmoid e tang. hip.
- 3 : DROPOUT - técnica proposta para evitar overfitting da rede
- 4 : Disponibilidade de Dados para treino - aumentou consideravelmente nas ultimas décadas (internet, smartphones, redes sociais, etc.).

Todas essas aplicações possuem dados rotulados (ou seja, imagens c

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN)

- Introdução

- Inspiração biológica
- Características da rede
- Exemplo ilustrativo
- Rede LeNet

3 Funções Utilizadas

Convolutional Neural Networks (CNN)

- Variantes biologicamente inspiradas das MLPs.
- Através do trabalho de Hubel e Wiesel acerca do córtex visual de gatos, sabe-se que o córtex visual é dotado de um complexo arranjo de células.
- Essas células, por sua vez, apresentam um **campo receptivo**, composto por pequenas sub-regiões que cobrem todo o campo visual.
 - Agem como **filtros locais** sobre o espaço de entrada, podendo explorar correlações espacialmente locais presentes em imagens.

Inspiração biológica

- Foram identificados dois tipos de células:
 - Células simples: respondem melhor a padrões semelhantes a fronteiras dentro de seu campo receptivo.
 - Células complexas: são dotadas de campo receptivo maior e localmente invariante à exata posição do padrão.
- CNNs reproduzem esse mecanismo.

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN)

- Introdução

- Inspiração biológica
- **Características da rede**
- Exemplo ilustrativo
- Rede LeNet

3 Funções Utilizadas

Conectividade esparsa

- As entradas das unidades escondidas na camada m são provenientes de um subconjunto das unidades da camada $m - 1$, composto por unidades com campos receptivos espacialmente contíguos.

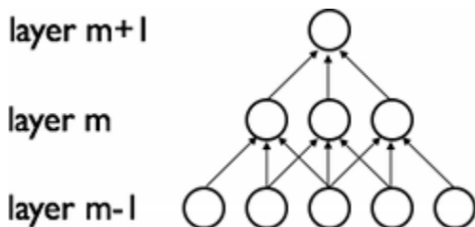


Figura 5: Ilustração de CNN.

Conectividade esparsa

- Na Figura 5, unidades da camada m apresentam campo receptivo de largura 3. Portanto, conectadas a apenas três neurônios da camada $m - 1$.
- Unidades da camada $m + 1$ têm conectividade semelhante com a camada m , contudo seu campo receptivo com relação à camada $m - 1$ é 5.
- Empilhar diversas camadas como essas leva a filtros não-lineares cada vez mais globais.
- CNNs podem ser treinadas por uma versão modificada do gradiente descendente estocástico.

Pesos compartilhados

- Cada filtro h_i é replicado através de todo o campo visual.
 - Essas unidades replicadas compartilham a mesma parametrização, e formam um **mapa de características (feature map)**.

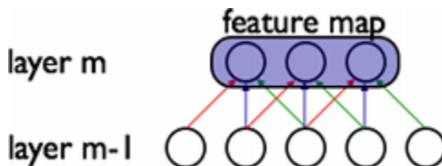


Figura 6: Neste exemplo, três unidades escondidas pertencem a um mesmo mapa de características (em destaque). Pesos da mesma cor são compartilhados (i.e. restritos a serem idênticos).

Pesos compartilhados

- Replicar unidades dessa maneira permite que características sejam detectadas independentemente de sua posição no campo visual.
 - Boa generalização em problemas de visão computacional.
- Além disso, o número de parâmetros livres a serem aprendidos é reduzido.

Detalhes e notação

- Um mapa de características é obtido aplicando repetidamente uma função sobre sub-regiões da imagem completa.
 - Isto é, realizando **convolução** da imagem de entrada com um classificador linear, adicionando um termo bias e aplicando uma função não-linear.
- Sendo h_k o k -ésimo mapa de características de uma dada camada, cujos filtros são determinados pelos pesos W^k e bias b_k , então o mapa h_k é dado por:

$$h_{ij}^k = \tanh \left((W^k * x)_{ij} + b_k \right)$$

- Para uma representação mais rica dos dados, cada camada escondida é composta por múltiplos mapas de características, isto é, $\{h^{(k)}, k = 0, \dots, K\}$.

Detalhes e notação

- Os pesos W de uma camada escondida são representados por um tensor tetra-dimensional, contendo elementos para cada:
 - *Feature map* de destino;
 - *Feature map* de origem;
 - Posição vertical de origem;
 - Posição horizontal de origem.
- Os bias b podem ser representados como um vetor contendo um elemento para cada mapa de características de destino.

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN)

- Introdução

- Inspiração biológica
- Características da rede
- **Exemplo ilustrativo**
- Rede LeNet

3 Funções Utilizadas

Exemplo ilustrativo

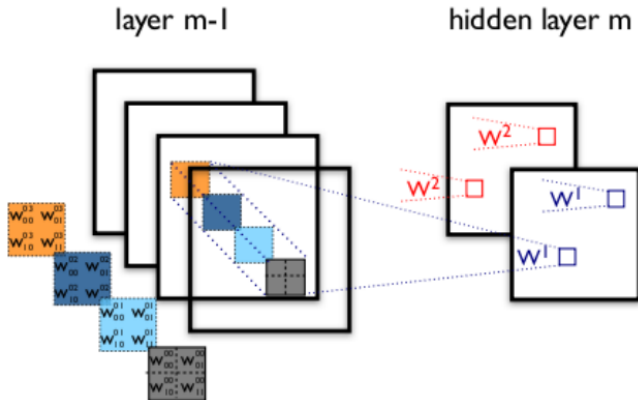


Figura 7: Exemplo de camada convolucional.

Camada Convolutiva

- Um neurônio numa cam. conv. é conectado a neurônios apenas numa região pequena da cam. anterior 5×5 , 7×7 ou 11×11 .
- O tamanho deste grid é chamado de *filtro*; uma cam conv. realiza uma convolução das suas entradas.
- Imita o padrão visto em áreas perceptuais do cérebro, células ganglionares retinais ou do córtex visual primário.
- Numa cam. conv., os pesos dos filtros são os mesmos para cada neurônio naquela camada;

Camada Convolutiva - Cont.

- Geralmente, uma cam. conv. é implementada como muitas "sub-layers" cada uma com um filtro diferente.
- Centenas de diferentes filtros de suas entradas podem ser usados em 1 cam. conv.
- Os resultados destas convulação são disponibilizados para a prox. cam.

Exemplo ilustrativo

- A Figura 7 mostra duas camadas de uma CNN.
- A camada $m - 1$ contém quatro mapas de características, enquanto a camada oculta m contém dois mapas de características (h_0 e h_1).
- Pixels (saídas dos neurônios) em h_0 e h_1 , representados como quadrados vermelhos e azuis, são computados a partir de pixels da camada $m - 1$, contidos dentro de seus campos receptivos 2×2 , representados como quadrados coloridos.

Exemplo ilustrativo

- Note que o campo receptivo abrange os quatro mapas de características de entrada.
 - Os pesos W_0 e W_1 de h_0 e h_1 são, portanto, tensores tridimensionais.
- W_{ij}^{kl} denota o peso conectando cada pixel do k -ésimo mapa de características da camada m ao pixel com coordenadas (i, j) do l -ésimo mapa de características da camada $m - 1$.

Sumário

1 Introdução

2 Convolutional Neural Networks (CNN)

- Introdução

- Inspiração biológica
- Características da rede
- Exemplo ilustrativo
- Rede LeNet

3 Funções Utilizadas

Redes LeNet

- Arquitetura proposta por Yann LeCun.
- Famosa família de redes convolucionais, com bons resultados na classificação de dígitos escritos com caligrafia manual.
- **GoogLeNet:** modelo usado no Google Deep Dream.
 - Código-fonte, em Python, é livre e está disponível no GitHub.
- A seguir, será apresentada a rede descrita em LeCun [1990].

LeNet Le Cun et al. [1990]

- Reconhecimento de dígitos numéricos, manuais ou impressos.
- **Base de dados:** digitalizações de códigos postais provenientes de correspondências verdadeiras do serviço postal de Buffalo - NY, suplementadas por dígitos impressos com diferentes fontes.
- Dez unidades na camada de saída, correspondentes a cada um dos dígitos que podem ser reconhecidos (classes).
 - Quando um padrão pertencente à classe i é apresentado, a saída é $+1$ para a i -ésima unidade de saída, e -1 para todas as outras.
- Rede não é totalmente conectada \rightarrow menos hiper-parâmetros a serem otimizados.
- O treinamento é realizado com uso do algoritmo *backpropagation*.

LeNet Le Cun et al. [1990]

- Somente conexões locais nas primeiras camadas.
 - Reconhecimento de formas (*shape recognition*) por meio de detecção de combinações de características locais.
- Se um detector de características é útil em uma parte da imagem, é provável que seja útil, também, em outras partes.
 - Características salientes de um caracter distorcido podem estar ligeiramente deslocadas da posição que teriam em um caracter típico.
- **Solução:** processar toda a camada de entrada por um único neurônio, com campo receptivo local, e armazenar os estados desse neurônio nas localizações correspondentes em um **mapa de características**.
 - Essa operação é equivalente a uma **convolução** com *kernel* pequeno, seguida de uma **função sigmóide**.

LeNet Le Cun et al. [1990]

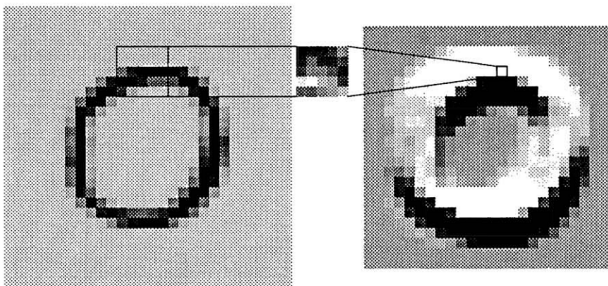


Figura 8: Imagem de entrada (esquerda), vetor de pesos (centro) e mapa de características resultante (direita). O mapa de características é obtido processando a imagem de entrada com um único neurônio com campo receptivo local, conforme indicado.

LeNet Le Cun et al. [1990]

- O processo pode ser executado em paralelo, implementando o mapa de características como um plano de neurônios cujos vetores de pesos são restritos a serem iguais (equivale a enxergar o processo como se houvesse um único neurônio).
 - Em outras palavras, as unidades de um mapa de características devem realizar a mesma operação em diferentes partes da imagem.
- A ideia de mapas de características locais e convolucionais pode ser aplicada a múltiplas camadas ocultas, extraindo características cada vez mais complexas e abstratas.

LeNet Le Cun et al. [1990]

- Cada extração de características é seguida de uma camada adicional, que realiza *averaging* local e *subsampling*, a fim de reduzir a resolução do mapa de características.
 - Essa camada introduz certo nível de invariância a distorções e translações.
- Um módulo funcional na rede LeNet consiste em uma camada de mapas de características com pesos compartilhados e uma camada de *averaging* / *subsampling*.

LeNet Le Cun et al. [1990]

- A rede é composta por quatro camadas ocultas, denominadas H1, H2, H3 e H4.
 - H1 e H3 são extratores de características com pesos compartilhados.
 - H2 e H4 são camadas de averaging e subsampling.

LeNet Le Cun et al. [1990]

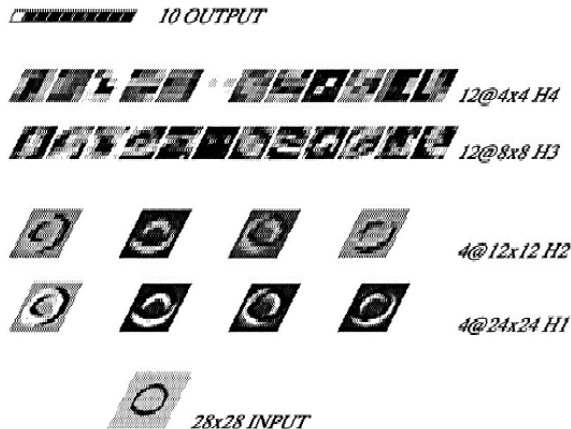


Figura 9: Arquitetura: 5 camadas de conexões totalmente adaptativas.

LeNet Le Cun et al. [1990]

- **H1:**

- Recebe um plano 28x28 como entrada.
- Contém quatro grupos de 576 unidades, organizadas em quatro mapas de características independentes com resolução 24x24, com sobreposição de 1 pixel.
 - H1.1, H1.2, H1.3 e H1.4
- Cada unidade em um mapa de características recebe, como entrada, uma vizinhança 5x5 do plano de entrada.
- Todas as 576 unidades em H1.1 usam o mesmo conjunto de 26 pesos (incluindo o bias), e o mesmo é válido para H1.2, H1.3 e H1.4.
 - Cada mapa tem seu próprio conjunto de pesos compartilhados.

LeNet Le Cun et al. [1990]

- **H2:**
 - Camada de *averaging/subsampling*.
 - Composta por quatro planos de resolução 12x12.
 - Cada unidade em cada um desses planos recebe entradas de quatro unidades do plano correspondente em H1.
 - Campos receptivos não se sobrepõem.
 - Pesos são restritos a serem iguais.

LeNet Le Cun et al. [1990]

- **H3:**
 - Composta por 12 mapas de características.
 - Cada mapa de características contém 64 unidades, arranjadas em um plano 8x8.
 - H3.1, H3.2, ..., H3.12.
 - O esquema de conexão entre H2 e H3 é semelhante ao que existe entre a entrada e H1, porém mais complicado.

LeNet Le Cun et al. [1990]

- **H3 (continuação):**

- Cada unidade no campo receptivo é composta por duas vizinhanças 5x5 centradas ao redor de unidades com posições idênticas em cada mapa de H2.
 - Pesos devem ser compartilhados entre as unidades de um dado mapa.
- Os mapas de H2 das quais H3 toma suas entradas são escolhidas de acordo com o esquema da Figura 10.

LeNet Le Cun et al. [1990]

	1	2	3	4	5	6	7	8	9	10	11	12
1	X	X	X		X	X						
2		X	X	X	X	X						
3							X	X	X		X	X
4								X	X	X		X

Figura 10: Conexões entre H2 e H3.

LeNet Le Cun et al. [1990]

- **H4:**
 - Tem o mesmo papel de H2.
 - Composta por 12 grupos de 16 unidades, arranjadas em planos 4x4.
- **Saída:** tem 10 unidades e é totalmente conectada a H4.
- A rede possui 4635 neuronios, 98442 pesos e 2578 parametros independentes.

LeNet Le Cun et al. [1990]

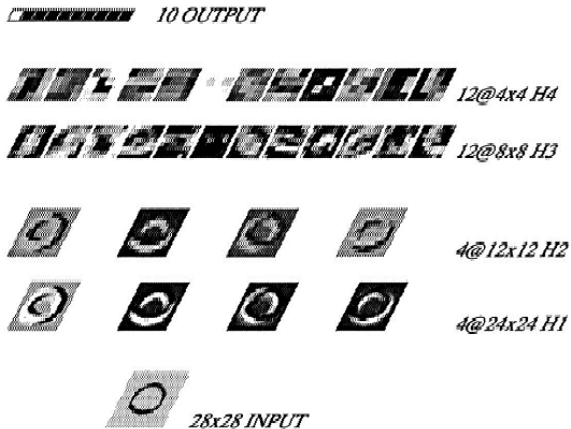


Figura 11: Arquitetura: 5 camadas de conexões totalmente adaptativas.

LeNet Le Cun et al. [1990]

- Conjunto de treinamento: 7291 caracteres manuscritos + 2549 dígitos impressos.
- Depois de 30 ciclos de treinamento a taxa de erro no conjunto de treinamento foi de 1.1% e o MSE foi de 0.017.
- Conjunto de teste: 2007 caracteres manuscritos + 700 dígitos impressos.
- Precisão no conjunto de teste foi de 3.4% e o MSE foi de 0.024

Sumário

1 Introdução

2 Convolutional Neural
Networks (CNN)

3 Funções Utilizadas

- Função Softmax
- Função ReLu
- Normalização: Dropout

Sumário

- 1 Introdução
- 2 Convolutional Neural Networks (CNN)
- 3 Funções Utilizadas**
 - **Função Softmax**
 - Função ReLu
 - Normalização: Dropout

Problema

- Em redes neurais *feedforward*, o aprendizado pode tornar-se lento ou pouco efetivo, devido a alguns fatores
- A otimização da função *erro quadrático médio* provoca saturação dos neurônios de saída, tornando o aprendizado mais lento.
- Quando uma entrada x_j é próxima de zero, o peso correspondente w_j se ajusta muito devagar.
- Otimizar a função *cross-entropy*, em vez do erro quadrático médio.
- Usar a função de ativação *softmax* nos neurônios de saída.

Função cross-entropy

- Pode ser a função a ser otimizada, alternativamente ao erro quadrático médio.

$$\text{cross-entropy} = - \sum y' \log y$$

Função Softmax

- A função *softmax* pode ser utilizada como função de ativação na última camada de uma rede neural.
- Seja v_i a combinação linear das entradas e pesos do neurônio de saída i , define-se:

$$\text{softmax}(v_i) = \frac{\exp(v_i)}{\sum_j \exp(v_j)}$$

- Aplicando a função *softmax* na camada de saída, obtém-se uma **distribuição de probabilidade** válida, em que a soma das saídas dos neurônios de saída é igual a 1.
 - A saída de cada neurônio representa a probabilidade de a classe correspondente ser a classificação correta.

Sumário

- 1 Introdução
- 2 Convolutional Neural Networks (CNN)
- 3 Funções Utilizadas**
 - Função Softmax
 - Função ReLu**
 - Normalização: Dropout

Função ReLu

- A função ReLu, ou **rectified linear unit**, é definida por:

$$f(x) = x^+ = \max\{0, x\}$$

- Uma aproximação suave é a função analítica **softplus**, dada por:

$$f(x) = \log[1 + \exp(x)]$$

Função ReLu

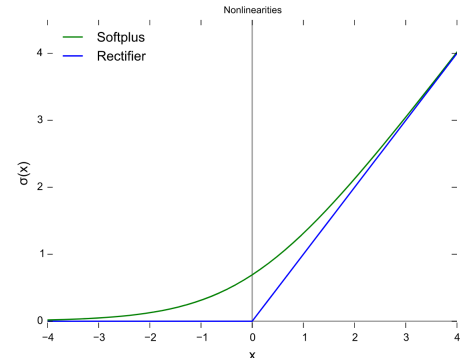


Figura 12: Funções ReLu (azul) e softplus (verde), próximas a $x=0$.

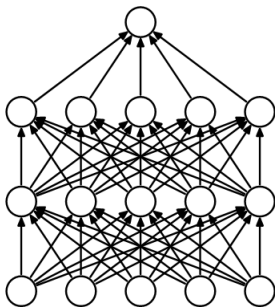
Sumário

- 1 Introdução
- 2 Convolutional Neural Networks (CNN)
- 3 Funções Utilizadas
 - Função Softmax
 - Função ReLu
 - Normalização: Dropout

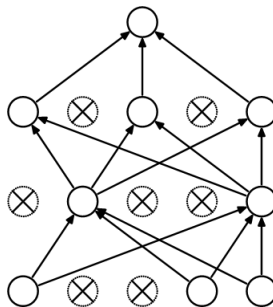
Dropout

- Em uma rede neural grande, muitas das conexões podem armazenar ruídos dos dados de entrada, levando a *Overfitting*.
- O termo *dropout* se refere a temporariamente remover unidades de uma rede neural, gerando uma rede diluída (*thinned*).
 - A escolha das unidades a serem removidas é, tipicamente, aleatória.
 - A probabilidade de uma unidade ser removida pode ser determinada em um conjunto de validação, ou simplesmente configurada como $p = 0,5$.
 - Para unidades de entrada, a probabilidade ótima costuma ser mais próxima de 1 do que de 0,5.

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Figura 13: À esquerda, rede neural comum, com duas camadas intermediárias. À direita, rede neural diluída (*thinned*), produzida com aplicação de *dropout*.

Dropout

- Durante o treinamento, é produzida uma rede diluída diferente para cada iteração.
- Durante o teste ou a aplicação da rede, todos os neurônios devem estar presentes.

Dropout

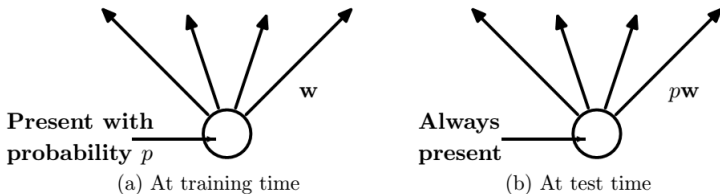


Figura 14: Uma unidade, durante o treinamento, está presente com probabilidade p . Durante o teste ou aplicação da rede, está sempre presente.

Passo a Passo dos Filtros

<https://towardsdatascience.com/cutting-edge-face-recognition-is-complicated-these-spreadsheets-make-it-easier-e7864dbf0e1a>

Referências I

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Le Cun, B. B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Minsky, M. and Papert, S. (1969). *Perceptrons*. MIT press.