Inteligência Artificial

Deep Learning

Prof. Saulo Popov Zambiasi
saulopz@gmail.com

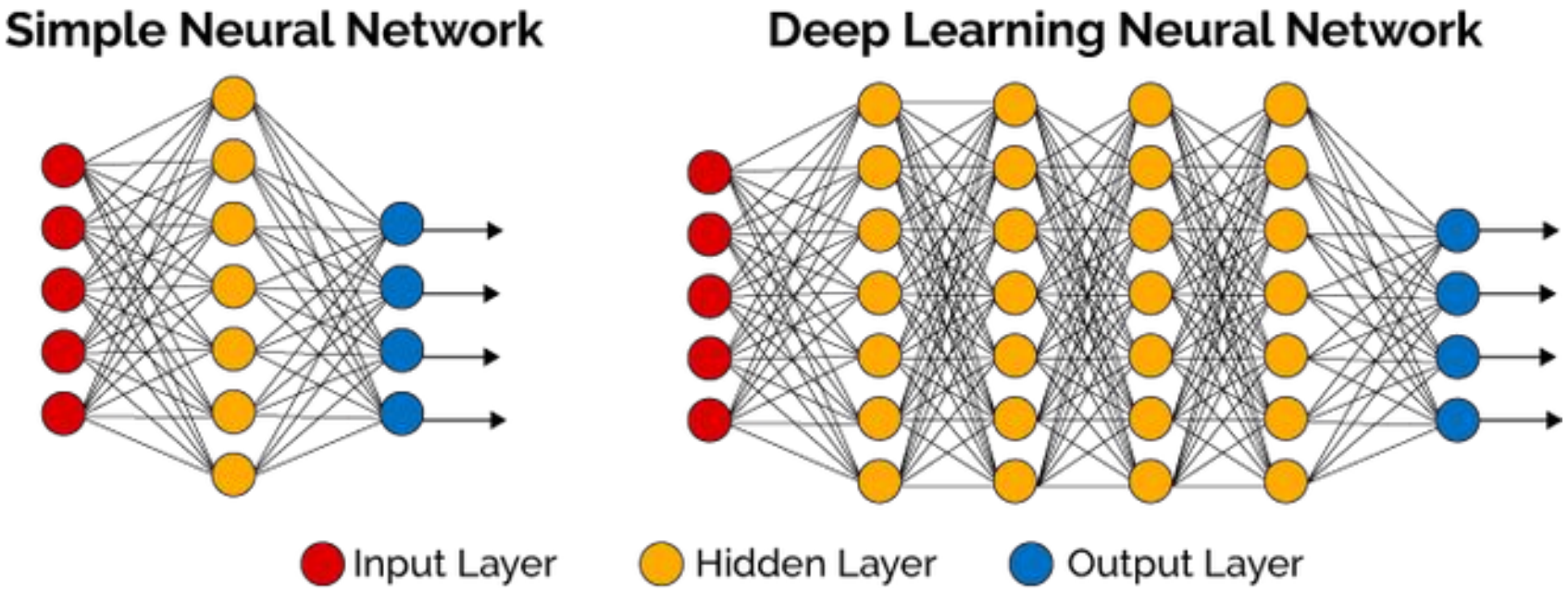
Aprendizagem Profunda – *Deep Learing*

- Subárea da Aprendizagem de Máquina.
- Usa algoritmos para processar dados e imitar o processamento feito pelo cérebro humano.
- Evolução das Redes Neurais Artificiais (são Redes Neurais Artificiais Profundas).
- Usa camadas de neurônios artificiais para
  - processar dados
  - compreender a fala humana
  - reconhecer objetos visualmente

Aprendizagem Profunda – *Deep Learing*

- A informação é passada através de cada camada, com a saída da camada anterior fornecendo entrada para a próxima camada.
- A primeira camada da rede é a camada de entrada e a última é a camada de saída.
- Todas as camadas entre as duas são as camadas ocultas.
- Cada camada é tipicamente um algoritmo simples e uniforme contendo um tipo de função de ativação.

Deep Learning



Deep Learning

- Responsável por avanços recentes em
  - visão computacional
  - reconhecimento de fala
  - processamento de linguagem natural
  - reconhecimento de áudio
- Utiliza **extração de recursos**: algoritmo para construir automaticamente “recursos” significativos dos dados para fins de treinamento, aprendizado e compreensão.
  - o Cientista de Dados, ou Engenheiro de IA, é responsável pela extração de recursos.
- Grande interesse atual da mídia popular e revistas científicas alavancando pesquisas e utilização de *Deep Learning*.

Neocognitron

- As primeiras **redes neurais convolucionais** foram usadas por Kunihiko Fukushima.
  - Redes neurais com múltiplas camadas de agrupamento e convoluções.
- Fukushima desenvolveu em 1979 uma rede neural chamada Neocognitron
  - design hierárquico e multicamadas.
  - permitiu ao computador aprender a reconhecer padrões visuais.
  - Se assemelhava a versões modernas, mas foram treinadas com uma estratégia de reforço de ativação recorrente em múltiplas camadas
  - permitiu que os recursos importantes fossem ajustados manualmente aumentando o peso de certas conexões

Fukushima

Neocognitron

- Conceitos de Neocognitron continuam a ser utilizados.
- Uso de conexões de cima para baixo e novos métodos de aprendizagem permitiram a realização de uma variedade de redes neurais.
- Quando mais de um padrão é apresentado ao mesmo tempo, o Modelo de Atenção Seletiva pode separar e reconhecer padrões individuais deslocando sua atenção de um para o outro.
- Um Neocognitron moderno não só pode identificar padrões com informações faltantes (por exemplo, um número 5 desenhado de maneira incompleta), mas também pode completar a imagem adicionando as informações que faltam.
  - Processo chamado de **inferência**.

Entra o Backpropagation

- O uso de erros no treinamento de modelos de Deep Learning, evoluiu significativamente em 1970 com o **Backpropagation**.
- Mestrado de Seppo Linnainmaa com o Backpropagation implementado em FORTRAN.

Linnainmaa

Entra o Backpropagation

- Infelizmente, o conceito não foi aplicado às redes neurais até 1985 quando Rumelhart, Williams e Hinton demonstraram o Backpropagation em uma rede neural que poderia fornecer representações de distribuição “interessantes”.


David Rumelhart


Ronald J. Williams


Geoffrey E. Hilton

Entra o Backpropagation

- Filosoficamente
  - essa descoberta trouxe à luz a questão dentro da psicologia cognitiva de saber se a compreensão humana depende da lógica simbólica (computacionalismo) ou de representações distribuídas (conexão).
- Yann LeCun (1989) – primeira demonstração prática de Backpropagation no Bell Labs
  - combinou redes neurais convolutivas com Backpropagation para ler os dígitos “manuscritos”

Yann LeCun

Inverno da IA

- 1985-1990 – Inverno da IA: afetou pesquisas em Redes Neurais e Aprendizagem Profunda.
- Expectativas otimistas frustraram o potencial da IA, irritando os investidores.
- A a frase Inteligência Artificial atingiu o status de pseudociência.

Inverno da IA

- 1995 – Corinna Cortes e Vladimir Vapnik desenvolveram a máquina de vetor de suporte ou ***Support Vector Machine***
  - sistema para mapear e reconhecer dados semelhantes.


Corinna Cortes


Vladimir Vaptnik


Sepp Hochreiter


Juergen Schmidhuber

- 1997 – Sepp Hochreiter e Juergen Schmidhuber desenvolveram o LSTM (*Long-Short Term Memory*) para redes neurais recorrentes.

Um novo passo evolutivo

- 1999 – Um novo passo evolutivo significativo para Deep Learning
- Computadores mais rápidos
- GPUs (unidades de processamento de gráfico)
  - um salto no tempo de processamento
  - aumento das velocidades computacionais em 1000 vezes ao longo de um período de 10 anos.
- As RNAs começaram a competir com máquinas de vetor de suporte.
  - RNA mais lenta em comparação com uma máquina de vetor de suporte
  - mas ofereciam melhores resultados usando os mesmos dados.
  - RNAs também têm a vantagem de continuar a melhorar à medida que mais dados de treinamento são adicionados.

Problema do *Vanishing Gradient* – 2000

- Características aprendidas em camadas mais baixas não eram aprendidas pelas camadas superiores, pois nenhum sinal de aprendizado alcançava essas camadas.
- Problema fundamental para RNAs com métodos de aprendizagem baseados em gradientes.
- Funções de ativação condensavam sua entrada, reduzindo a faixa de saída de forma caótica.
- Produzia grandes áreas de entrada mapeadas em uma faixa extremamente pequena.
- Nessas áreas de entrada, uma grande mudança era reduzida a uma pequena mudança na saída, resultando em um gradiente em queda.
- Duas soluções utilizadas para resolver este problema foram
  - pré-treino camada-a-camada
  - desenvolvimento de uma memória longa e de curto prazo.

Ainda no histórico

- 2001 – Grupo META (atual Gartner), pesquisa sobre
  - desafios e oportunidades no crescimento do volume de dados
  - aumento do volume de dados e a crescente velocidade de dados como o aumento da gama de fontes e tipos de dados
  - Foi um pontapé para o Big Data, que estava apenas começando.

Ainda no histórico

Fei-Fei Li

- 2009 – Fei-Fei Li, professora de IA em Stanford na Califórnia
  - lançou o ImageNet
  - montou uma base de dados gratuita de mais de 14 milhões de imagens etiquetadas
  - Eram necessárias imagens marcadas para “treinar” as redes neurais.
  - Ela disse: “Nossa visão é que o Big Data mudará a maneira como a aprendizagem de máquina funciona. Data drives learning.”… e acertou em cheio!

Ainda no histórico

- Até 2011:
  - a velocidade das GPUs aumentou significativamente
  - possibilitou a formação de redes neurais convolutivas “sem” o pré-treino camada por camada
  - o aumento da velocidade de computação trouxe vantagens significativas ao Deep Learning
  - Exemplo, AlexNet, ConvNet, cuja arquitetura ganhou várias competições internacionais durante 2011 e 2012

Ainda no histórico

- 2012, o Google Brain lançou os resultados de um projeto incomum conhecido como The Cat Experiment
  - explorava as dificuldades de “aprendizagem sem supervisão” usado na aprendizagem profunda (uso de dados rotulados)
  - Usando a aprendizagem sem supervisão, uma rede neural convolucional é alimentada com dados não marcados, e é então solicitada a busca de padrões recorrentes

Google Brain

Cat Experiment

- Rede neural distribuída por mais de 1.000 computadores
- Dez milhões de imagens “sem etiqueta” foram tiradas aleatoriamente do YouTube, mostradas ao sistema e, em seguida, o software de treinamento foi autorizado a ser executado
- No final do treinamento, um neurônio na camada mais alta foi encontrado para responder fortemente às imagens de gatos.
- Andrew Ng, o fundador do projeto, disse: “**Nós também encontramos um neurônio que respondeu fortemente aos rostos humanos**”.
- A aprendizagem não supervisionada continua a ser um um campo ativo de pesquisa em Aprendizagem Profunda.

Andrew Ng

Ainda no histórico

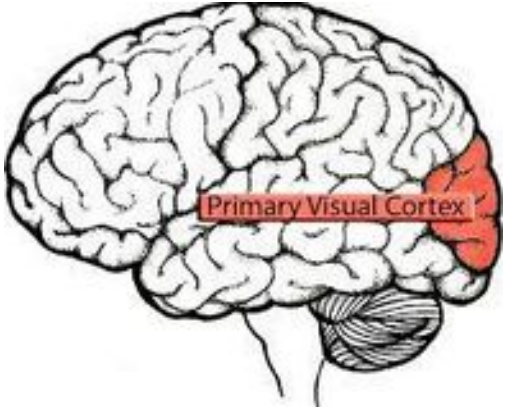
- Atualmente, o processamento de Big Data e a evolução da Inteligência Artificial são ambos dependentes da Aprendizagem Profunda.
- Com Deep Learning podemos construir sistemas inteligentes e estamos nos aproximando da criação de uma IA totalmente autônoma.
- Isso vai gerar impacto em todas os segmentos da sociedade e aqueles que souberem trabalhar com a tecnologia, serão os líderes desse novo mundo que se apresenta diante de nós.

Redes Neurais Convolucionais

- Rede Neural Convolucional / ConvNet / *Convolutional Neural Network* / CNN
- Algoritmo de Aprendizado Profundo
- Pega uma imagem de entrada, atribui importância (pesos e vieses que podem ser aprendidos) a vários aspectos / objetos da imagem e ser capaz de diferenciar um do outro
- Pré-processamento muito menor em comparação com outros algoritmos de classificação.

Redes Neurais Convolucionais

- Métodos primitivos os filtros são feitos à mão, com treinamento suficiente, mas as ConvNets têm a capacidade de aprender esses filtros / características.
- Arquitetura
  - análoga a do padrão de conectividade de neurônios no cérebro humano
  - inspirada na organização do Visual Cortex
  - neurônios individuais respondem a estímulos apenas em uma região restrita do campo visual conhecida como Campo Receptivo
  - uma coleção desses campos se sobrepõe para cobrir toda a área visual



ConvNet

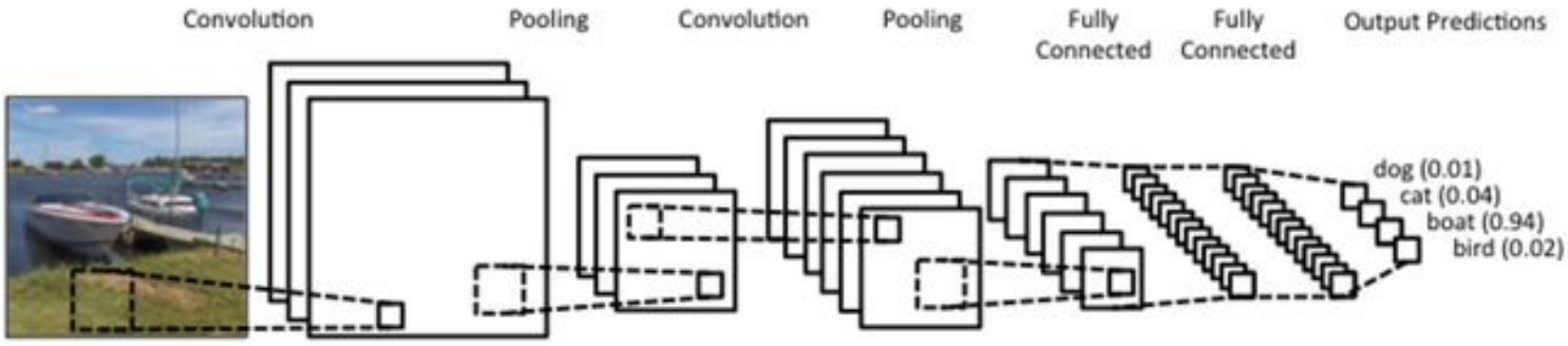
- Uma imagem é apenas uma matriz de valores de pixels, certo?
- Então, por que não apenas achatar a imagem (por exemplo, converter uma matriz 3×3 em um vetor 9×1. Se a image é uma matriz, nenhum problema em converter em uma vetor) e alimentá-lo para um Perceptron Multi-Layer para fins de classificação?

Na verdade não!

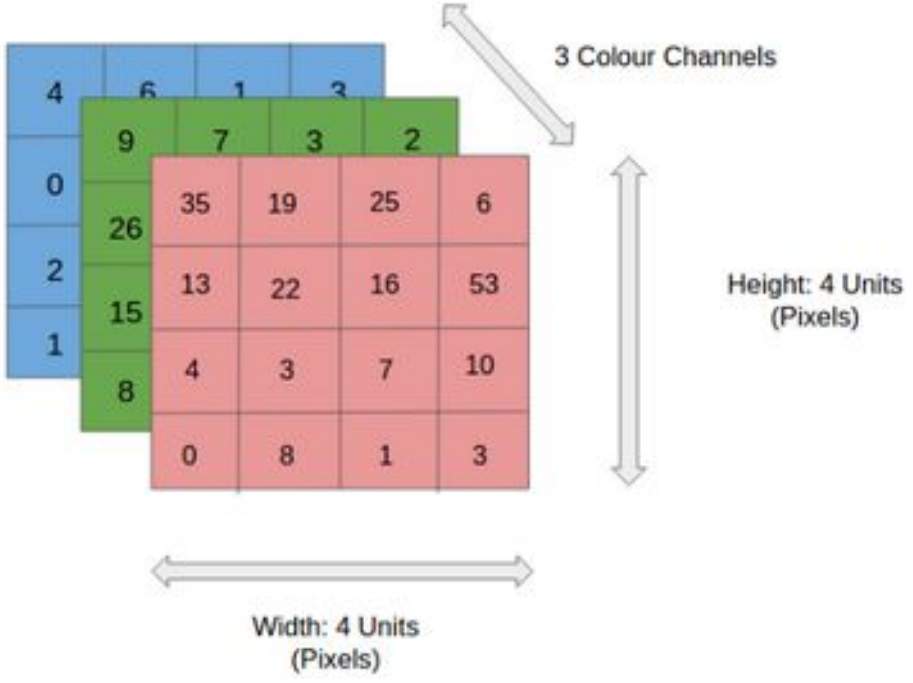
- No caso de imagens binárias extremamente básicas, o método pode mostrar uma pontuação de precisão média durante a previsão de classes, mas teria pouca ou nenhuma precisão quando se trata de imagens complexas com dependências de pixel por toda parte

ConvNet

- Uma ConvNet é capaz de capturar com sucesso as dependências espaciais e temporais em uma imagem através da aplicação de filtros relevantes.
- A arquitetura executa um melhor ajuste ao conjunto de dados da imagem devido à redução no número de parâmetros envolvidos e à capacidade de reutilização dos pesos.
- Em outras palavras, a rede pode ser treinada para entender melhor a sofisticação da imagem.



ConvNet



- Na figura vemos uma imagem RGB separada pelos três planos coloridos
- Os cálculos ficariam pesados demais com imagens de dimenções de 8K, por exemplo.
- A função da ConvNet é reduzir as imagens para uma forma mais fácil de processar, sem perder recursos que são críticos para obter uma boa previsão.
- Isso é importante quando queremos projetar uma arquitetura que não seja apenas boa em recursos de aprendizado, mas que também seja escalável para conjuntos de dados massivos.

ConvNet

- ConvNets usam três ideias básicas:
  - campos receptivos locais
  - pesos compartilhados
  - pooling

Campos Receptivos Locais

- Visão Computacional
- Uma pessoa jogar uma bola pra você pegar é uma tarefa fácil, certo?
  - **Errado**. É um processos muito complexos
  - Como o cérebro processa a visão de modo que sabemos exatamente o que é uma bola e quando ela está vindo em nossa direção?
  - Ensinar uma máquina que seja capaz de ver da mesma forma que nós seres humanos é uma tarefa realmente difícil.

Campos Receptivos Locais

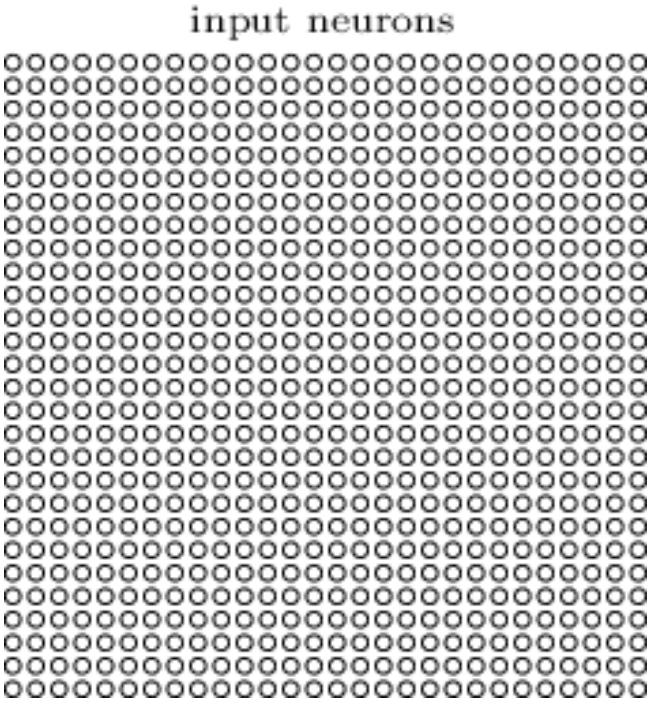
- Resumindo de forma aproximada o processo
  - A imagem da esfera passa através de seu olho e chega a sua retina
  - Ocorre alguma análise elementar e envia o resultado ao cérebro
  - O córtex visual analisa mais profundamente a imagem
  - Ele envia para o resto do córtex, que compara a tudo o que já sabe, classifica os objetos e dimensões
  - Decide sobre algo a fazer: levantar a mão e pegar a bola (tendo previsto o seu caminho).
  - Isso ocorre em uma pequena fração de segundo, com quase nenhum esforço consciente e quase nunca falha.
  - Recriar a visão humana não é apenas um problema difícil, é um conjunto deles, cada um dos quais depende do outro.

Campos Receptivos Locais

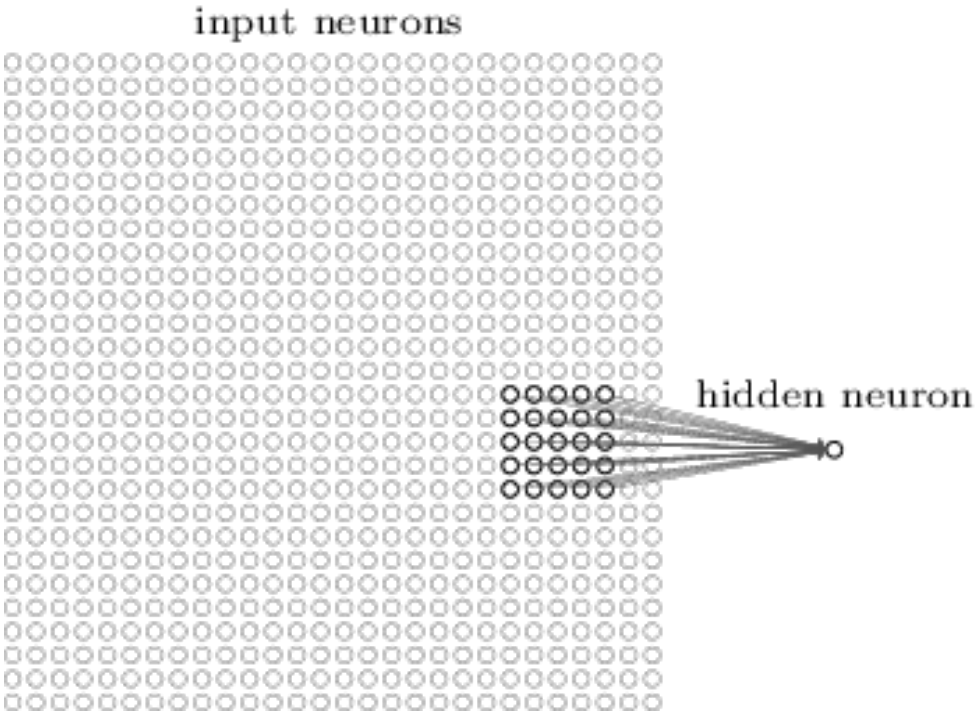
- **Visão Computacional**
  - processo de modelagem e replicação da visão humana usando software e hardware.
  - disciplina que estuda como reconstruir, interromper e compreender uma cena 3d a partir de suas imagens 2d em termos das propriedades da estrutura presente na cena.
- Visão Computacional e reconhecimento de imagem são termos frequentemente usados como sinônimos, mas o primeiro abrange mais do que apenas analisar imagens.
- Mesmo para os seres humanos, “ver” também envolve a percepção em muitas outras frentes, juntamente com uma série de análises.
- Uma pessoa usa cerca de dois terços do seu cérebro para o processamento visual
  - por isso não é nenhuma surpresa que os computadores precisariam usar mais do que apenas o reconhecimento de imagem para obter sua visão de forma correta.

Campos Receptivos Locais

- Para compreender o que é um campo receptivo local, vamos considerar a imagem de seu formato padrão de 28x28.
- Ccada pixel é um valor numérico que representa a intensidade de cor de acordo com a escala de cor utilizada, como RGB (Red – Green – Blue), por exemplo, ou apenas intensidade em escala de cinza para imagens em preto e branco.



Campos Receptivos Locais

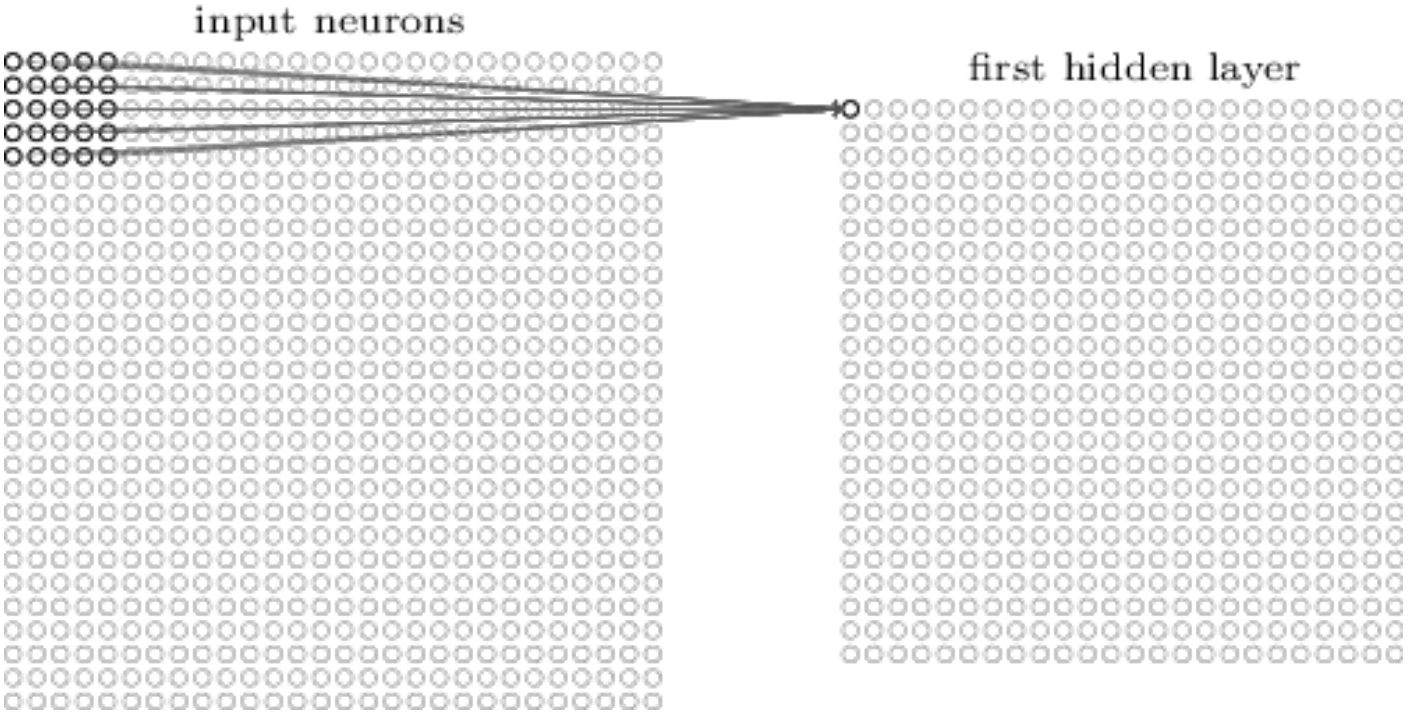


- Conectamos os pixels de entrada a uma camada de neurônios ocultos.
- Mas não vamos conectar todos os pixels de entrada a cada neurônio oculto.
- Faremos apenas conexões em regiões pequenas e localizadas da imagem de entrada.
- Ou seja, cada neurônio na primeira camada oculta será conectado a uma pequena região dos neurônios de entrada, por exemplo, uma região de 5 × 5, correspondendo a 25 pixels de entrada.
- Assim, para um neurônio oculto em particular, podemos ter conexões que se parecem com a imagem ao lado

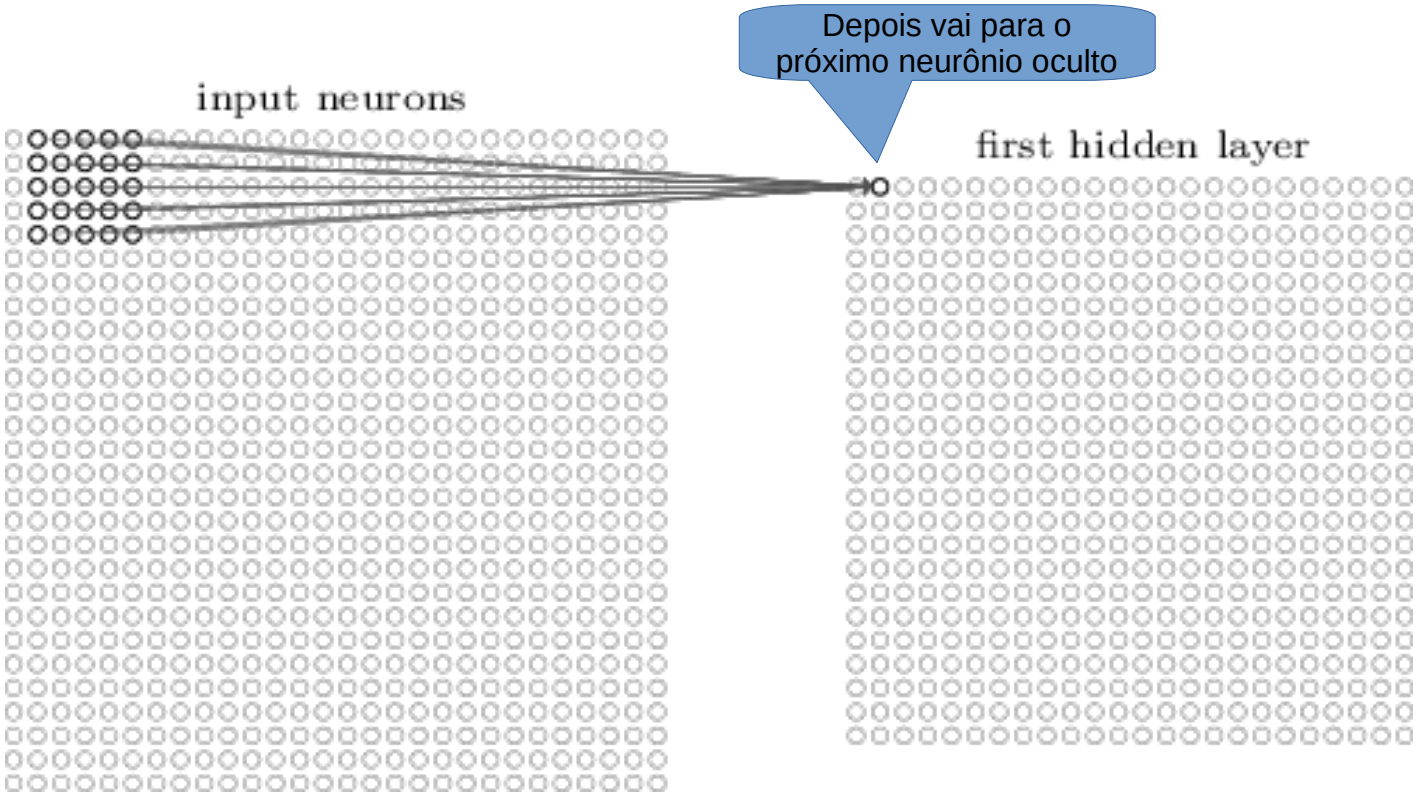
Campos Receptivos Locais

- Essa região na imagem de entrada é chamada de campo receptivo local para o neurônio oculto.
- É uma pequena janela nos pixels de entrada.
- Cada conexão aprende um peso e o neurônio oculto também aprende um viés (bias) geral.
- O bias é uma forma de analisar seu campo receptivo local específico.
- Depois deslizamos o campo receptivo local por toda a imagem de entrada.
- Para cada campo receptivo local, existe um neurônio oculto diferente na primeira camada oculta.
- Iniciamos com um campo receptivo local no canto superior esquerdo

Campos Receptivos Locais



Campos Receptivos Locais



Campos Receptivos Locais

- E assim por diante, construindo a primeira camada oculta.
- Com uma imagem de entrada 28 × 28 e campos receptivos locais 5 × 5, haverá 24 × 24 neurônios na camada oculta.
- Isso ocorre porque só podemos mover o campo receptivo local 23 neurônios para o lado (ou 23 neurônios para baixo), antes de colidir com o lado direito (ou inferior) da imagem de entrada.
- Mostramos o campo receptivo local sendo movido por um pixel por vez. Contudo, às vezes, um comprimento de passada diferente é usado.
  - Por exemplo, podemos mover o campo receptivo local 2 pixels para a direita (ou para baixo), caso em que diríamos que um comprimento de passada de 2 é usado.
  - Esse é um dos hiperparâmetros de uma rede neural convolucional, chamado *stride length*.
  - No exemplo visto é usado um *stride length* de 1, mas vale a pena saber que as pessoas às vezes experimentam comprimentos de passada diferentes.

Pesos Compartilhados

- Cada neurônio nosso tem um viés e pesos 5 × 5 conectados ao seu campo receptivo local.
- Iremos usar os mesmos pesos e vieses para cada um dos 24 × 24 neurônios ocultos.
- Assim, para o neurônio oculto, a saída é:

$$\sigma\left(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right)$$

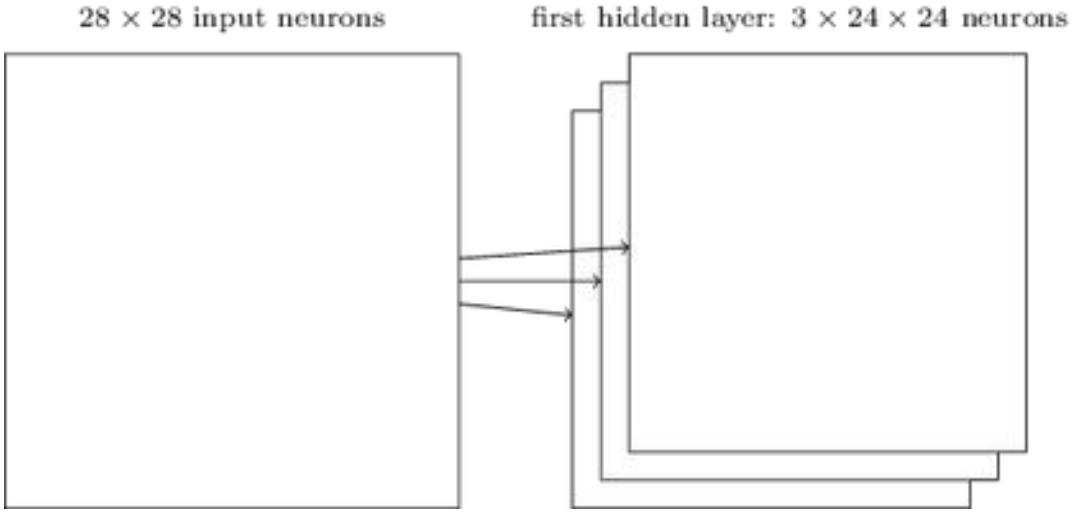
- Assim, todos os neurônios da primeira camada oculta detectam exatamente o mesmo recurso, apenas em locais diferentes na imagem de entrada.

Pesos Compartilhados

- Porque isso faz sentido?
  - suponha que os pesos e os vieses sejam tais que o neurônio oculto possa escolher, por exemplo, uma borda vertical em um campo receptivo local específico.
  - Essa habilidade também é útil em outros lugares da imagem.
  - Por isso, é útil aplicar o mesmo detector de recursos em toda a imagem.
- Para colocar esse conceito em termos um pouco mais abstratos, as redes convolucionais são bem adaptadas à invariância da translação das imagens:
  - girar uma foto de um gato 90 graus, ainda faz dela a imagem de um gato, embora os pixels agora estejam organizados de forma diferente.

Pesos Compartilhados

- A estrutura de rede descrita pode detectar apenas um único tipo de recurso localizado.
- Para fazer reconhecimento de imagem, precisamos de mais de um mapa de recursos.
- Uma camada convolucional completa consiste em vários mapas de recursos, diferentes:

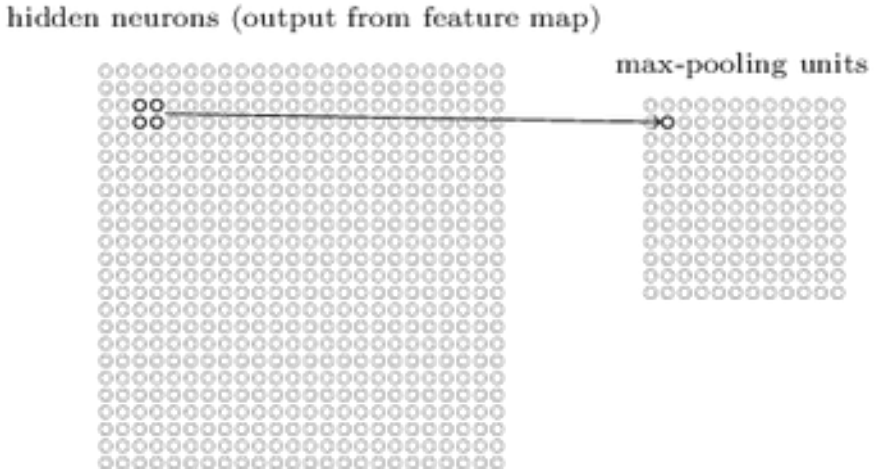


Pesos Compartilhados

- Agora existem 3 mapas de recursos.
  - Cada mapa de recursos é definido por um conjunto de pesos compartilhados de 5 × 5 e um único viés compartilhado.
  - O resultado é que a rede pode detectar três tipos diferentes de recursos, sendo cada recurso detectável em toda a imagem.
- No exemplo temos apenas 3 mapas de recursos, para manter o diagrama acima simples. No entanto, na prática, as redes convolucionais podem usar mais (e talvez muito mais) mapas de recursos.

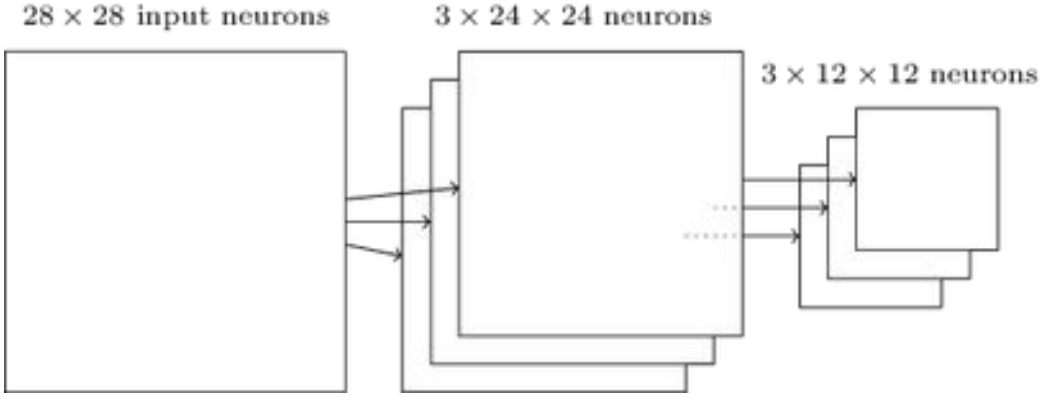
Camadas de Pooling

- Uma camada de pooling recebe cada saída do mapa de características da camada convolucional e prepara um mapa de características condensadas.
- Por exemplo, cada unidade na camada de pooling pode resumir uma região de 2 × 2 neurônios na camada anterior.
- Como um exemplo concreto, um procedimento comum para o pooling é conhecido como pool máximo (ou Max-Pooling).
- No Max-Pooling, uma unidade de pooling simplesmente gera a ativação máxima na região de entrada 2 × 2, conforme ilustrado no diagrama a seguir



Camadas de Pooling

- Note que, como temos 24 × 24 neurônios emitidos da camada convolucional, após o agrupamento, temos 12 × 12 neurônios.
- Aplicamos o Max-Pooling para cada mapa de recursos separadamente. Portanto, se houvesse três mapas de recursos, as camadas combinadas, convolutional e Max-Pooling, se pareceriam com a imagem a seguir.

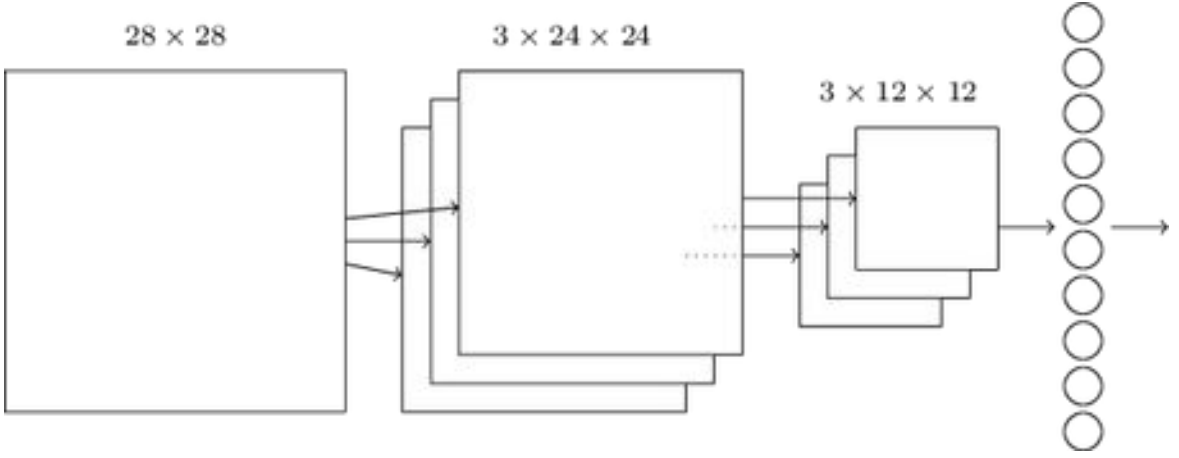


Canadas de Pooling

- O Max-Pooling pode ser visto como uma forma de a rede perguntar se um determinado recurso é encontrado em qualquer lugar de uma região da imagem.
- Em seguida, elimina a informação posicional exata.
- A ideia é que uma vez que um recurso tenha sido encontrado, sua localização exata não é tão importante quanto sua localização aproximada em relação a outros recursos.
- Um grande benefício é que há muito menos recursos agrupados e, portanto, isso ajuda a reduzir o número de parâmetros necessários nas camadas posteriores.
- O Max-Pooling não é a única técnica usada para o pooling. Outra abordagem comum é conhecida como Pooling L2.
  - Em vez de tomar a ativação máxima de uma região 2 × 2 de neurônios, tomamos a raiz quadrada da soma dos quadrados das ativações na região 2 × 2.

ConvNet

- Juntando tudo, temos uma ConvNet
- No exemplo, temos uma camada de 10 neurônios de saída, correspondentes aos 10 valores possíveis para dígitos MNIST (‘0’, ‘1’, ‘2’, etc)



ConvNet

- A rede começa com 28 × 28 neurônios de entrada
  - cada image de cada dígito do dataset MNIST tem 28 x 28 pixels
- Os neurônios são usados para codificar as intensidades de pixel para uma imagem no dataset MNIST.
- Segue-se uma camada convolucional usando um campo receptivo local de 5 x 5 e três mapas de características.
- O resultado é uma camada de 3 × 24 × 24 neurônios ocultos.
- A próxima etapa é uma camada de Max-Pooling, aplicada a regiões 2 × 2, em cada um dos três mapas de recursos.
- O resultado é uma camada de 3 × 12 × 12 neurônios ocultos.
- A camada final de conexões na rede é uma camada totalmente conectada.
  - essa camada conecta todos os neurônios da camada de max-pooling a cada um dos 10 neurônios de saída

Inteligência Artificial

Deep Learning

Prof. Saulo Popov Zambiasi
saulopz@gmail.com