Object Recognition with Gradient-Based Learning Yann Lecun

Nator e Pedro

December 12, 2023

Outline

Temáticas do Artigo

- **CNN como extrator de características:**
  - Modelos tradicionais de segmentação e extração de características;
  - Modelos principais de classificação;
  - Apresentação de CNNs como modelos de aprendizado adequados para reconhecimento de objetos.
  - LeNet-5;
  - Rede Neural de Deslocamento Espacial (SDNN)
- Apresentação de uma abordagem de aprendizado baseado em gradiente para modelos em **grafos**

Aprendendo os recursos certos

O modelo mais comumente aceito de reconhecimento de padrões é composto por três partes principais:

- **Segmentador:** O papel do segmentador é separar os objetos de interesse do fundo em que estão. Ele isola os objetos que queremos analisar do restante da imagem.
- **Extrator de Características:** O extrator de características é responsável por coletar informações relevantes a partir das imagens de entrada. Ele remove informações não relevantes e variações indesejadas, focando apenas no que é importante para a análise.
- **Classificador:** O classificador é a parte que coloca cada objeto em categorias específicas. Ele recebe as representações de características dos objetos (normalmente como vetores ou sequências de símbolos) e decide a que categoria cada objeto pertence.

Aprendendo os recursos certos

Existem três métodos principais para a classificação:

- **Correspondência de Modelo (Template Matching):** Compara a representação de características com um conjunto de modelos de classe predefinidos;
- **Métodos Generativos:** Usam modelos de densidade de probabilidade para cada classe e selecionam a classe com a maior probabilidade de gerar a representação de características;
- **Modelos Discriminativos:** Calculam uma função discriminante que atribui uma pontuação diretamente a cada classe.

Aprendendo os recursos certos

O segmentador e o extrator de características frequentemente **baseiam-se em suposições simplificadas** sobre os dados de entrada e raramente podem considerar todas as variações do mundo real.

- O pré-processamento mínimo garante que nenhuma suposição irrealista seja feita sobre os dados;
- Isso requer criar uma **arquitetura** de aprendizado **adequada** que possa lidar com a **alta dimensão** da entrada (número de pixels)

Reconhecimento de forma com CNNs

As redes neurais de várias camadas tradicionais, onde todas as unidades em **uma camada estão conectadas a todas as unidades na próxima camada**, podem ser usadas para reconhecer imagens "brutas" (aproximadamente normalizadas em tamanho e centralizadas), **mas existem problemas**.

- Imagens passam por **várias camadas** de cálculos para identificar padrões;
- Essas redes podem enfrentar desafios ao lidar com a **complexidade** e **variabilidade do mundo real**.
- **Dificuldade** em reconhecer objetos em várias **posições** ou **tamanhos**.
- Complexidade dos cálculos em cada camada pode resultar em **treinamento demorado** e **ajustes difíceis**.

Reconhecimento de forma com CNNs

- Arquiteturas totalmente conectadas ignoram a topologia da entrada (MLP).
- A ordem das variáveis de entrada não afeta o treinamento (MLP).
- Imagens têm uma forte estrutura local 2D.
- Variáveis (pixels) próximas espacialmente têm alta correlação.
- Correlações locais são úteis para extrair e combinar características locais.
- Isso é vantajoso para reconhecer objetos espaciais ou temporais.
- Redes Convolucionais impõem a extração de características locais.

Rede Neural Convolucional

- **Conexões Locais e Campos Receptivos:** Cada unidade em uma camada se conecta a unidades em uma vizinhança local na camada anterior.
- **Extração de Características Elementares:** Unidades com campos receptivos locais podem aprender a detectar características visuais simples, como bordas e cantos, em uma imagem.
- **Combinação de Características:** As características extraídas pelas unidades com campos receptivos locais são combinadas em camadas subsequentes para detectar características mais complexas.
- **Variações de Posição:** Devido a distorções ou deslocamentos na entrada, as posições das características podem variar.
- **Detecção de Características em Toda a Imagem:** Detectores de características aprendidos em uma parte da imagem também podem ser úteis em outras partes.

Gradient-Based Learning

- O GBL fornece uma estrutura para construir um sistema.
- O aprendizado de máquina tem a função $Y^p = F(Z^p, W)$ em que Z^p é o p-th input W representa a coleção de parâmetros ajustáveis no sistema Y^p contém pontuações ou probabilidades para cada categoria

Função Erro

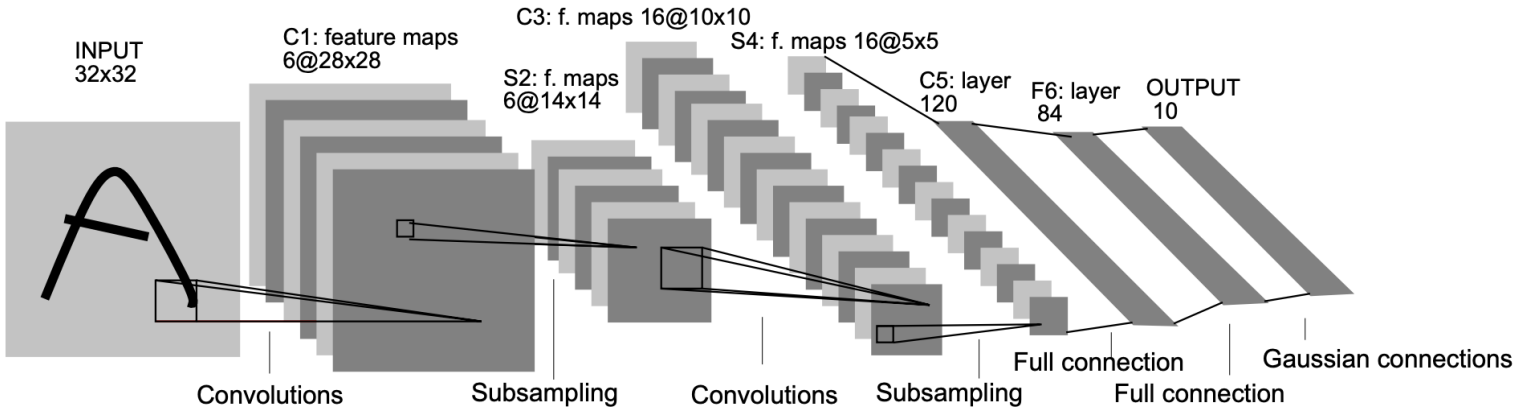
- A função erro $E^p = D(D^p, F(W, Z^p))$ mede a discrepância entre D^p da saída de Z^p e a saída produzida pelo sistema. A média da perda da função $E_{train}(W)$ é a média de erros E^p do conjunto de exemplos de treino $\{(Z^1, D^1), \ldots (Z^p, D^p)\}$. O aprendizado do problema consiste em encontrar o valor de W que minimize $E_{train}(W)$.
- A função $X_n = F_n(W_n, X_{n-1})$ em que X_n é um objeto representando a saída do módulo W_n é um vetor treinável, um subconjunto de W e X_{n-1} é um entrada de uma saída anterior da função. O X_0 é a primeira entrada dado pelo padrão de entrada Z^p.

backpropagation

- $\frac{\partial E^p}{\partial W_n} = \frac{\partial F_n}{\partial W}(W_n, X_{n-1})\frac{\partial E^p}{\partial X_n}$
- $\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F_n}{\partial X}(W_n, X_{n-1})\frac{\partial E^p}{\partial X_n}$
- temos $\frac{\partial F_n}{\partial W}(W_n, X_{n-1})$ que é uma Jacobiana de F_n em relação a W no ponto (W_n, X_{n-1}), e $\frac{\partial F_n}{\partial X}(W_n, X_{n-1})$ é uma Jacobiana de F_n em relação a X.

LeNet-5

- Uma típica rede convolucional para reconhecimento de formas chamada LeNet-5.



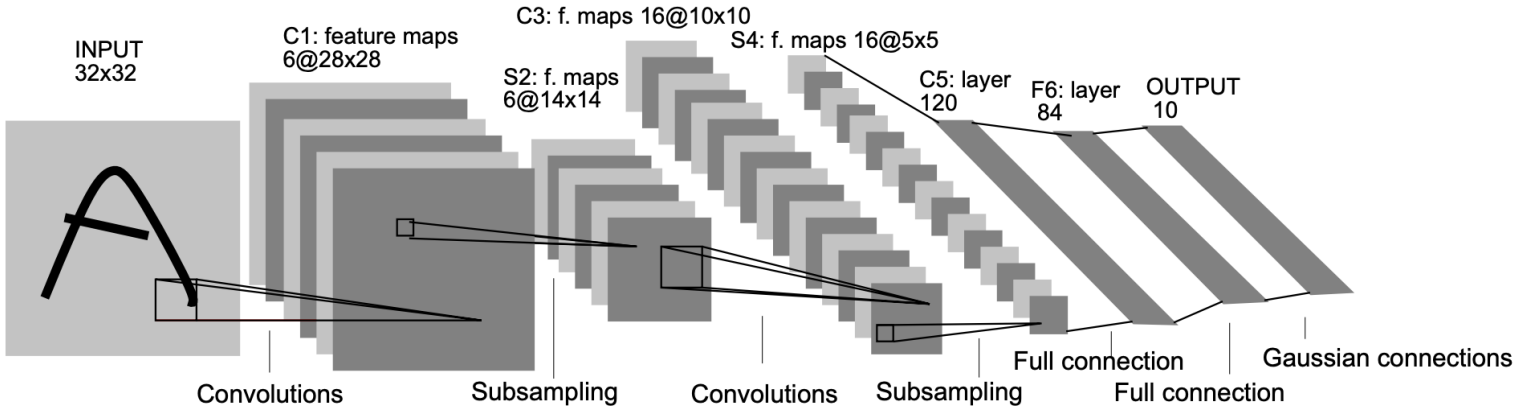
- Arquitetura da LeNet-5, uma CNN, que aqui reconhece dígitos. Cada plano é um mapa de características(feature map).

LeNet-5

- A LeNet-5 compreende 7 camadas, todos contendo parâmetros treináveis. A entrada é uma imagem 32x32 pixels.
- A primeira camada oculta de LeNet-5 é organizada em 6 planos, cada um com um feature map. Um feature map tem 25 entradas conectadas em um área de 5 por 5, chamada de campo receptivo (receptive field). Cada unidade tem 25 entradas, e portanto 25 coeficientes de treinamento mais um BIAS de treinamento.

LeNet-5

- A camada C1 é uma camada convolucional com 6 mapas de características. Cada unidade em cada mapa de características é conectada a uma matriz 5x5 da entrada. O tamanho do feature map é 28x28 evita que a conexão da entrada saia do limite. C1 contém 156 parâmetros de treinamento, e 122.304 conexões.

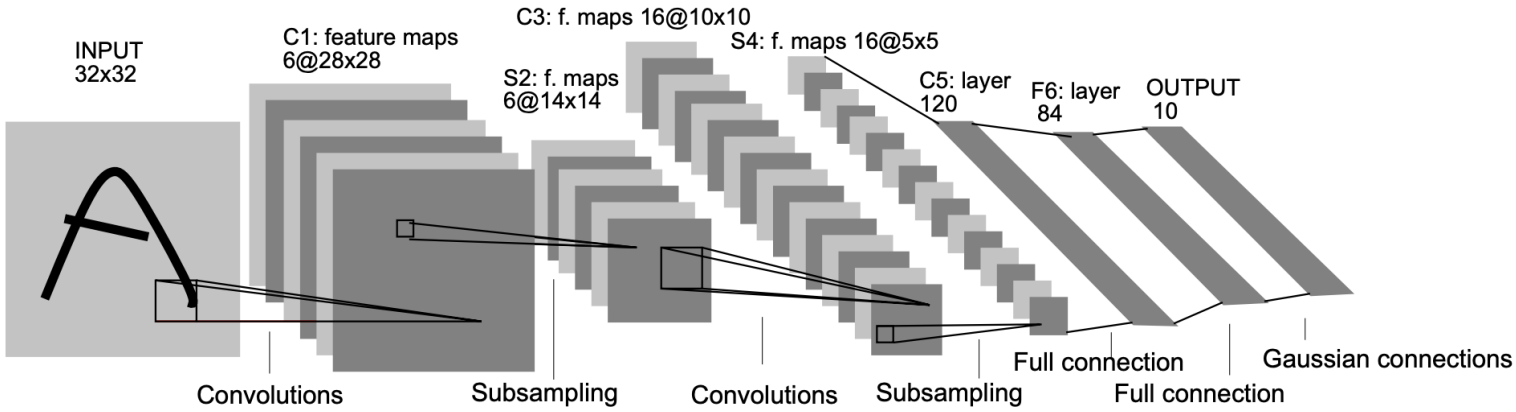


LeNet-5

- A camada S2 é uma camada subamostragem (subsampling layer) com 6 mapas de características de tamanho 14x14. Cada unidade em cada mapa de característica é conectada a um 2x2 correspondente ao mapa de características em C1. As quatro entradas de uma unidade em S2 são somadas, depois multiplicadas por um coeficiente treinável e adicionadas a um bias treinável. O resultado passa por uma função sigmóide. Com receptores de tamanho 2x2, portanto o mapa de características da camada S2 tem metade das linhas e colunas do mapa de características da camada C1. A camada S2 tem 12 parâmetros treináveis e 5880 conexões.

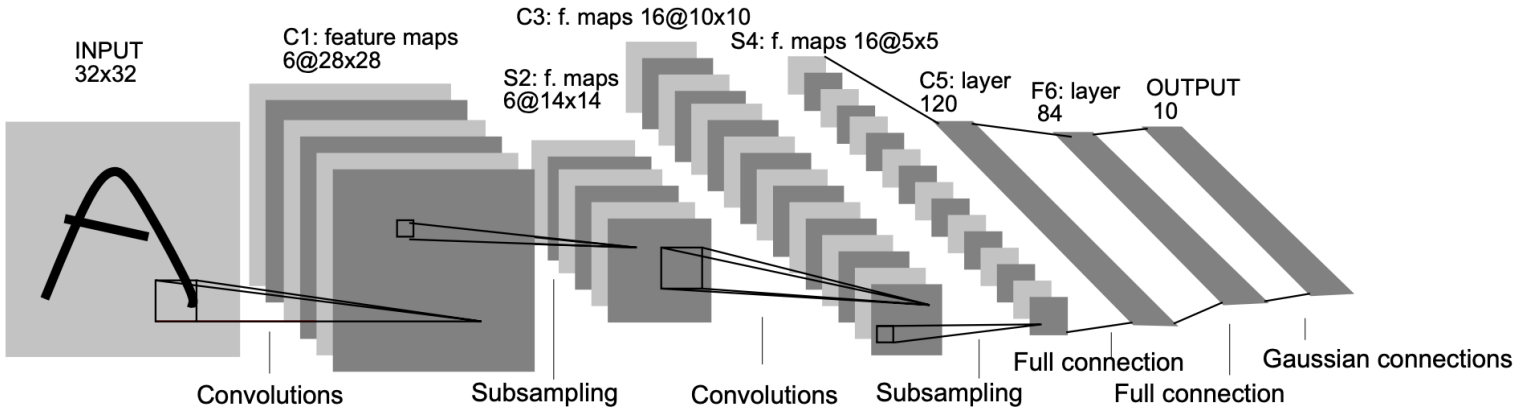
LeNet-5

- A camada C3 é uma camada de convolução com 16 mapas de características. Cada unidade em cada mapa de características é conectada a uma matriz 5x5 a um subconjunto do mapa de características de S2. A camada C3 tem 1516 parâmetros treináveis e 156.000 conexões.



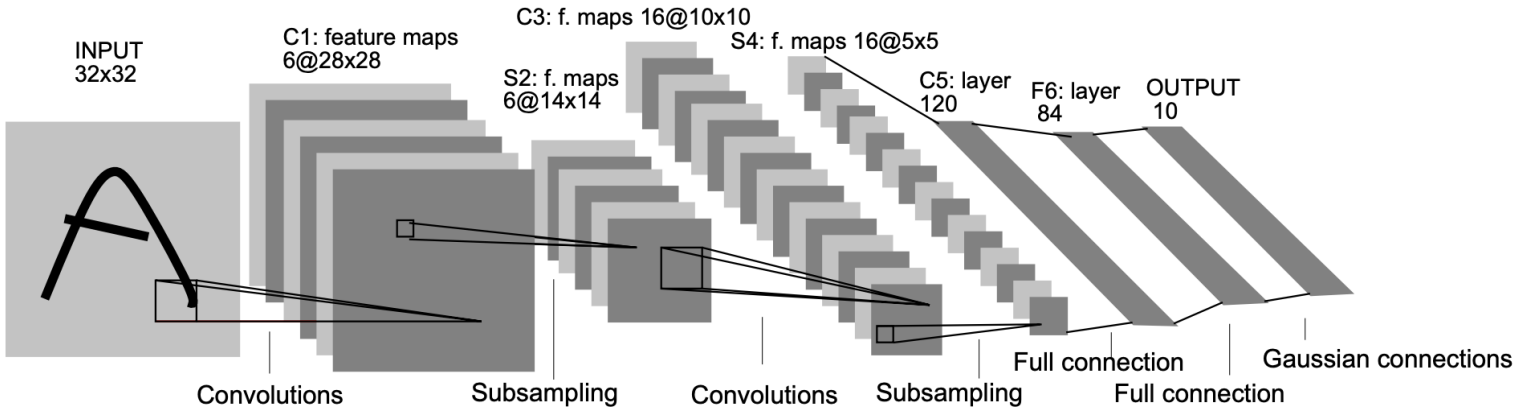
LeNet-5

- A camada S4 é um camada subamostral (sub-sampling layer) com 16 feature maps de tamanho 5x5. Cada unidade e cada feature map é conectado a um 2x2 correspondente do feature map de C3, similar a conexão entre C1 e S2. A camada S4 tem 32 parâmetros treináveis e 2000 conexões.



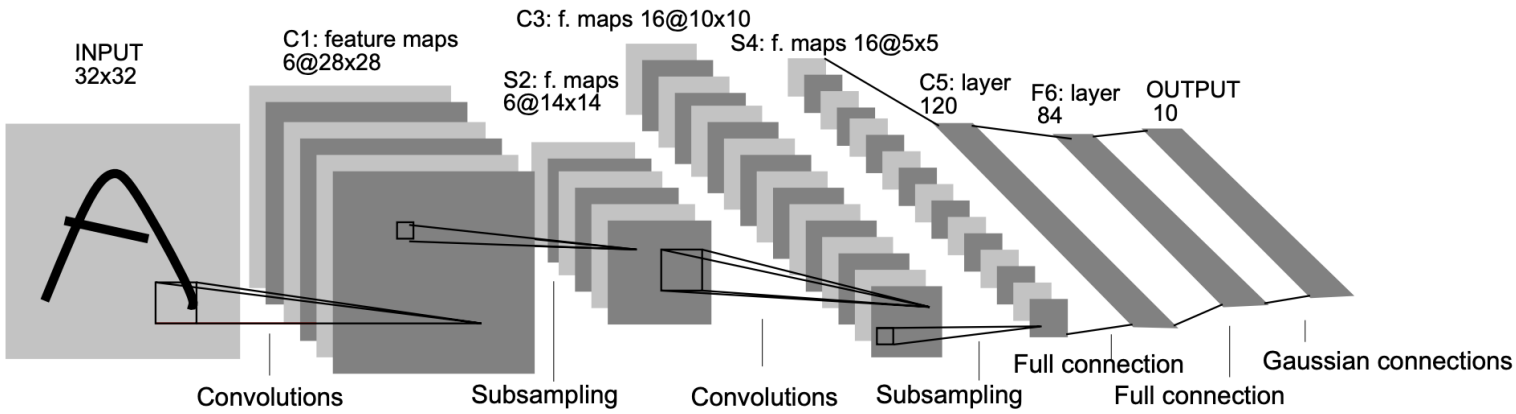
LeNet-5

- A camada C5 é uma camada convolucional com 120 features maps. Cada unidade é conectada a um 5x5 todos os 16 feature maps de S4. Aqui, por causa do tamanho do S4.



LeNet-5

- A camada F6, contém 84 unidades que é a unidade de saída composto por uma função de base radial euclidiana (Euclidean Radial Basis Function RBF).



LeNet-5

- A função perda da saída da rede é:
$$E(W) = \frac{1}{P} \sum_{p=1}^{P} yD^p(Z^p, W)$$
onde yD^p é a saída da Dp-ésima unidade RBF, ou seja, aquela que corresponde à classe correta do padrão de entrada Z^p.

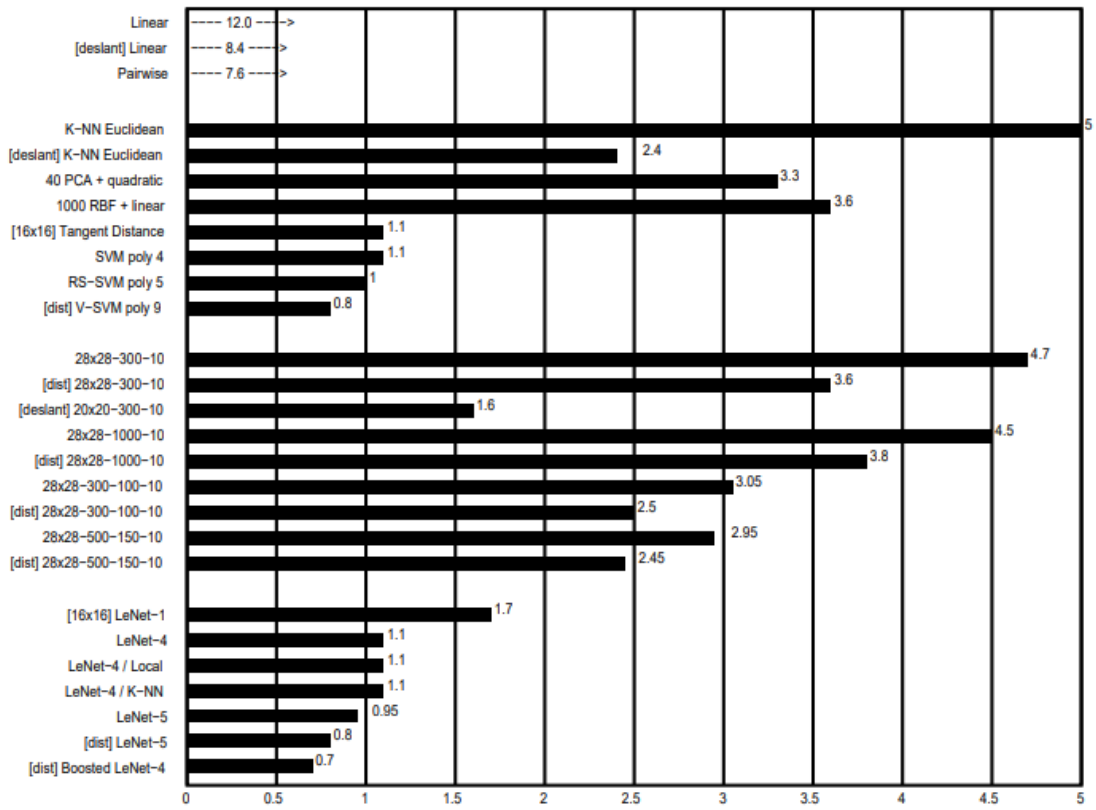
Exemplo

- O banco de dados usado para treinar e testar os sistemas descritos no artigo pertence a NIST's Special Database 3 and 1 contendo imagens binárias de dígitos manuscritos. O banco de dados chama-se MNIST contém 60.000 training samples (metade são SD1, metade são SD3), e 10.000 test images (metade são SD1 e metade são SD3).
- A entrada da imagem é normalizada em 20x20 pixels preservando o aspect ratio. O resultado é uma imagem em escala de cinza. Três versões do banco de dados é usado. A primeira versão, as imagens foram centralizadas em uma imagem 28x28 calculando o centro de massa dos pixels e transladando a imagem para este centro. Esta versão do banco de dados será chamada de banco de dados regular.

Exemplo

- Na segunda versão do banco de dados, (conhecida como versão delantada), as imagens dos caracteres foram desviadas usando os momentos de inércia dos pixels posteriores e cortadas em imagens de 20x20 pixels.
- Na terceira versão do banco de dados, utilizada em alguns experimentos iniciais, as imagens foram reduzidas para 16x16 pixels.

Resultado

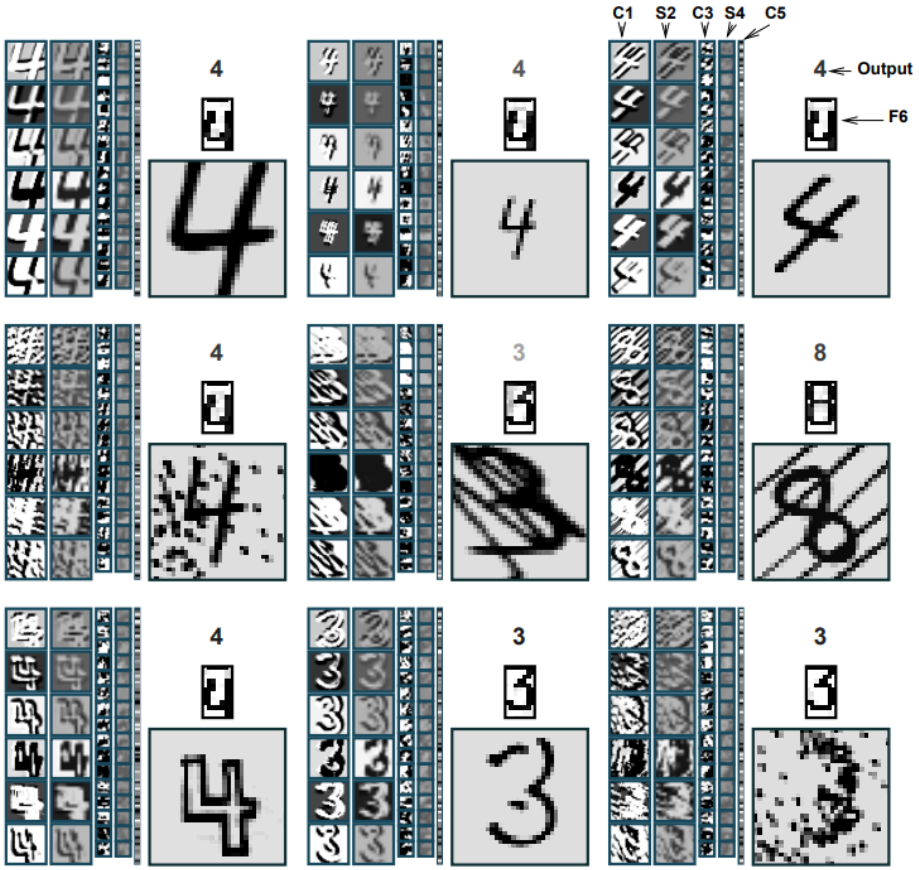


- taxa de erro no test set (%) para vários métodos de classificação. [deslant] indica que o classificador foi treinado e testado na versão deslante do banco de dados. [dist] indica que o training set tem argumentos aumentados com exemplos distorcidos artificialmente.

Evolução LeNet

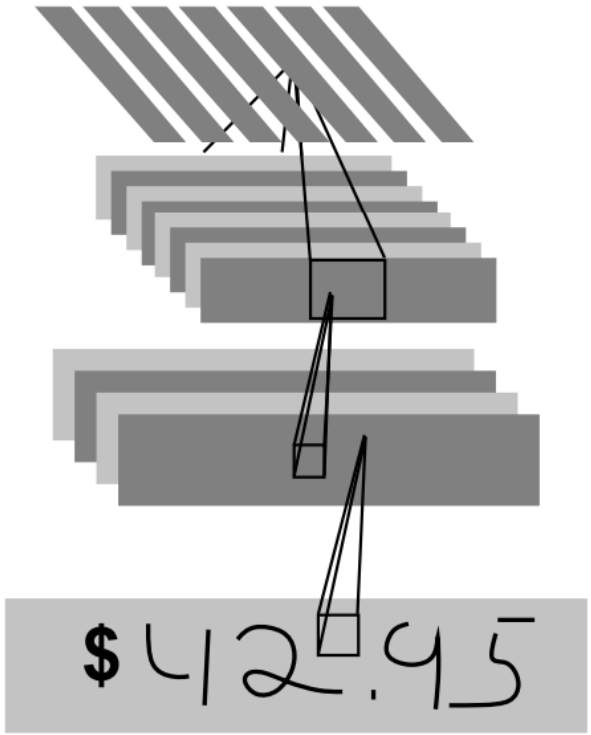
- LeNet-1 é uma CNN menor com apenas 2600 parâmetros livres e 100000 conexões.
- LeNet-4 é uma CNN com 17000 parâmetros livres e 260000 conexões.
- Boosted LeNet-4 é uma classificação que obteve a votação de três instâncias do LeNet-4 treinadas em diferentes subconjuntos do banco de dados.

Invariância e resistência ao ruído



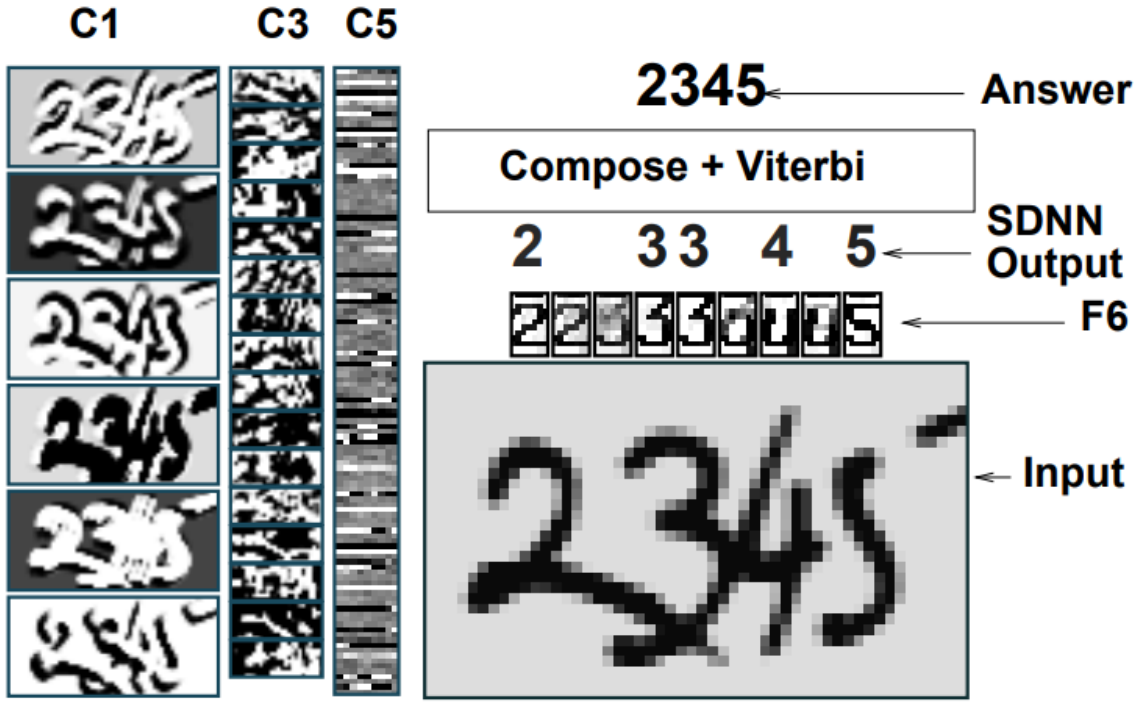
- Exemplos de distorção e ruído nos caracteres corrigidos pelo LeNet-5. O grau de cinza representa a penalidade da saída.

reconhecimento de múltiplos objetos com redes neurais de deslocamento espacial



- Uma rede neural de deslocamento espacial é uma rede convolucional que foi replicada em um amplo campo de entrada.

reconhecimento de múltiplos objetos com redes neurais de deslocamento espacial



- Um pós processamento é requerido para extrair a melhor sequência de saída.
- Viterbi é um algoritmo que obtêm uma estimativa de probabilidade máxima a posteriori na Estatísticas bayesianas.

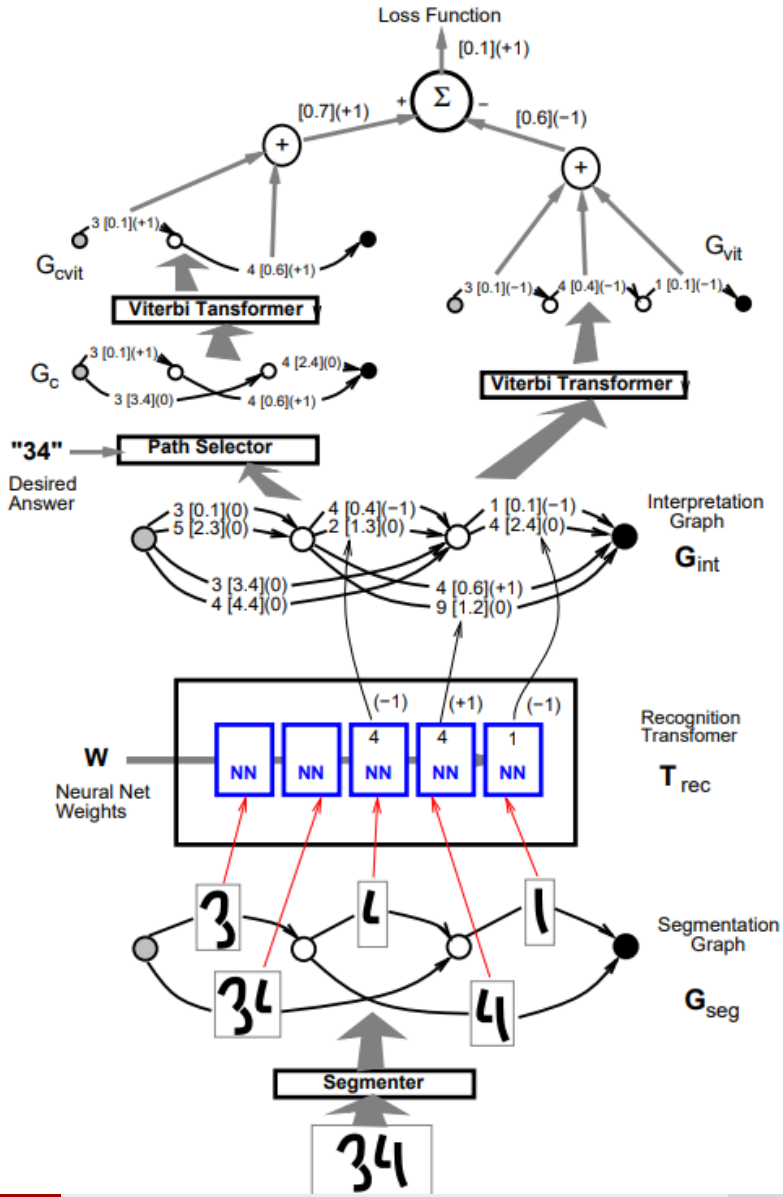
Detecção de rosto e avistamento com SDNN

- A principal ideia é treinar uma CNN para distinguir imagens do objeto de interesse das imagens presentes. A rede cobre toda a imagem a ser analisada, formando assim um SDNN bidimensional. A saída do SDNN é um plano bidimensional no qual as unidades mais ativas indicam a presença do objeto de interesse. A ideia é aplicada na localização facial do artigo An original approach for the localization of objects in images [Vaillant, Monrocq and LeCun 1994]

An original approach for the localization of objects in images

- Neste artigo referenciado, os atores utilizam a técnica mostrada até agora para trabalhar com imagens de tamanho 512*512, incluindo suavização e normalização da imagem. Este algoritmo facilita as operações é baseada em convoluções com kernels de tamanho 5*5 ou 8*8.
- Este tipo de algoritmo de segmentação pode ser aplicado a outros problemas onde os objetos a serem detectados não pode ser facilmente caracterizados pelo seu contorno ou pelos processamentos de imagens clássicas.

Graph Transformer Networks



Graph Transformer Networks

- Uma arquitetura GTN para reconhecimento de palavras baseada na super segmentação heurística. Durante o reconhecimento, apenas o caminho direito da parte superior é utilizado. Para treinamento com treinamento Viterbi, apenas o caminho da mão esquerda é usado. Para o treinamento discriminativo de Viterbi, ambos os caminhos são usados. As quantidades entre colchetes são penalidades calculadas durante a propagação direta. As quantidades entre parênteses são derivadas parciais calculadas durante a propagação para trás.