

Resumo Acadêmico: Fundamentos da Inteligência Artificial (Unidades 3 e 4)

Este documento sintetiza os conceitos centrais das Unidades 3 e 4 da apostila, abordando as arquiteturas de Aprendizado Profundo (Deep Learning) e os fundamentos da Visão Computacional.

Unidade 3: Introdução ao Deep Learning

A Unidade 3 explora o Deep Learning (Aprendizado Profundo) como uma evolução das Redes Neurais Artificiais (RNAs), caracterizada por arquiteturas com múltiplas camadas de processamento (profundidade). Essa abordagem é responsável pelos avanços recentes em IA, especialmente em Processamento de Linguagem Natural (PLN) e Visão Computacional.

Fundamentos das Redes Neurais Artificiais (RNAs)

- **Inspiração Biológica:** As RNAs são modelos matemáticos inspirados no sistema nervoso biológico, capazes de adquirir conhecimento através da aprendizagem.
- **Perceptron:** O primeiro modelo de RNA (proposto por Frank Rosenblatt, 1958), é um classificador linear binário. Sua limitação é a incapacidade de resolver problemas não linearmente separáveis (como a porta lógica XOR).
- **Perceptron Multicamadas (MLP):** Uma evolução que utiliza uma ou mais "camadas ocultas" (neurônios internos) entre a entrada e a saída.
 - **Função de Ativação:** Diferente da função "degrau" do Perceptron, o MLP utiliza funções não-lineares, como a **sigmoide**, que permite a representação de problemas complexos.
 - **Aprendizagem (Backpropagation):** O algoritmo de treinamento fundamental é o *error backpropagation* (retropropagação do erro). Ele calcula o erro na camada de saída e o propaga de trás para frente, ajustando os pesos das conexões em cada camada para minimizar o erro global.

Arquiteturas de Deep Learning

O que diferencia o Deep Learning das RNAs tradicionais é o uso de um número significativamente maior de camadas (dezenas ou centenas), permitindo a aprendizagem hierárquica de características.

1. Redes Neurais Convolucionais (CNNs - Convolutional Neural Networks):

- **Aplicação Principal:** Visão Computacional (processamento e classificação de imagens).
- **Conceito-chave: A convolução.** Em vez de conectar cada neurônio de entrada a cada neurônio da próxima camada (como no MLP), a CNN aplica "filtros" (kernels) que

deslizam sobre a imagem, detectando características específicas (bordas, texturas, formas).

- **Arquitetura Típica:**

- **Camada Convolucional:** Aplica os filtros para criar "mapas de características".
- **Camada de Retificação (ReLU):** Uma função de ativação $f(x) = \max(0, x)$ que introduz não-linearidade e acelera o treinamento.
- **Camada de Subamostragem (Pooling):** Reduz a dimensão dos mapas de características (ex: Max Pooling), tornando a rede mais eficiente e invariante a pequenas translações.
- **Camada Densa (Fully Connected):** Ao final, as características extraídas são "achatadas" (flattening) e passadas para um MLP tradicional para a classificação final.
- **Exemplos Notáveis:** LeNet-5 (reconhecimento de dígitos) e AlexNet (vencedora do ImageNet 2012, impulsionou o uso moderno de CNNs).

2. Redes Neurais Recorrentes (RNNs - Recurrent Neural Networks):

- **Aplicação Principal:** Processamento de dados sequenciais (PLN, séries temporais, reconhecimento de fala).
- **Conceito-chave:** "Memória". As RNNs possuem conexões que formam ciclos, permitindo que a informação de um passo de tempo anterior influencie o processamento do passo de tempo atual.
- **Problema (Vanishing Gradient):** Em sequências longas, o gradiente do erro pode "desaparecer" durante a retropropagação, dificultando o aprendizado de dependências de longo prazo.
- **LSTM (Long Short-Term Memory):** Uma arquitetura de RNN avançada que resolve o problema do desaparecimento do gradiente. A LSTM utiliza uma "célula de memória" e três "portões" (gates) — *keep gate*, *write gate* e *output gate* — que controlam rigorosamente quais informações devem ser esquecidas, mantidas ou adicionadas à memória da rede, permitindo o aprendizado de dependências muito longas.

Unidade 4: Introdução à Visão Computacional

A Unidade 4 detalha o campo interdisciplinar da Visão Computacional (VC), que visa capacitar computadores a "compreender" e extrair significado semântico de imagens e vídeos, simulando a visão humana.

Fundamentos e Utilizações

- **Evolução:** A VC evoluiu do Processamento de Imagens (década de 1970), passando por análises matemáticas 3D (décadas de 1980-1990), até a revolução do aprendizado de máquina e Deep Learning (pós-2010), impulsionada pelo poder das GPUs.
- **Aplicações na Indústria 4.0:** A VC é estratégica para a Indústria 4.0 e a Internet das Coisas (IoT), permitindo a automação de processos, controle de qualidade, carros autônomos e robótica avançada.
- **Tarefas Comuns:**
 1. **Reconhecimento e Localização:** Identifica um objeto (ex: "gato") e sua posição (através de uma *bounding box*).
 2. **Detecção de Objetos:** Localiza múltiplas instâncias de diferentes classes (ex: todos os "carros" e "pedestres" em uma cena).
 3. **Segmentação de Instâncias:** Tarefa mais complexa que delimita a fronteira exata, pixel a pixel, de cada objeto na imagem.

O Pipeline da Visão Computacional

A unidade detalha as etapas necessárias para construir um sistema de VC, seja pela abordagem clássica ou contemporânea.

1. Visão Computacional Clássica (Pré-Deep Learning):

- **Pré-processamento:** Prepara a imagem para análise (ex: redimensionamento, remoção de ruído, conversão para escala de cinza, translação, rotação).
- **Extração de Características (Manual):** Etapa crítica onde um especialista define e extraí manualmente padrões relevantes da imagem (ex: histogramas de cores, formatos, texturas) usando algoritmos como Haar Cascade (usado para detecção facial).
- **Treinamento e Avaliação:** As características extraídas (Vetor X) e seus rótulos (Vetor y) são usados para treinar um classificador de Machine Learning (ex: Scikit-Learn). O conjunto de dados é dividido em **treino, validação e teste** para evitar *overfitting* (quando o modelo decora os dados de treino mas não generaliza) ou *underfitting* (quando o modelo é simples demais).
- **Métricas de Avaliação:** Acurácia (percentual geral de acertos), Precisão (entre os positivos previstos, quantos estavam corretos) e Recall (entre todos os positivos reais, quantos foram encontrados).

2. Visão Computacional Contemporânea (Com Deep Learning):

- **Frameworks:** A complexidade do Deep Learning é abstraída por frameworks como **TensorFlow** (Google) e **Keras** (uma interface de alto nível que roda sobre o TensorFlow).

- **Extração de Características (Automática):** A principal vantagem das CNNs. A própria rede *aprende* quais são as características mais relevantes durante o treinamento, eliminando a necessidade da extração manual.
- **Avaliação do Modelo:** O monitoramento da **Acurácia** e da **Perda (Loss)** ao longo das épocas de treinamento, tanto nos dados de treino quanto nos de validação, é crucial para diagnosticar *overfitting* e ajustar hiperparâmetros (como a taxa de aprendizado do otimizador).