**Artigo:** Optimal Asset Allocation Using Adaptive Dynamic Programming   
**Autor:** Ralph Neuneier   
**Ano:** 1998

**Em geral:** A alocação de ativos é formalizada como um Problema Markoviano de decisão (MDP) e é otimizada com a aplicação, de forma comparativa, das técnicas de programação dinâmica e Reinforcement Learning (q-learning) .

Resultado preliminar: A estratégia de alocação utilizando Q-learning se mostrou equivalente à programação dinâmica

Os métodos são testados com a tarefa de investir capital líquido no mercado de ações alemão e redes neurais são utilizadas para aproximação de função de valores.

Objetivo: demonstrar a aplicabilidade da técnica de aprendizado por reforço baseado em redes neurais à um problema com espaço de estados de alta dimensão.

**Método e dados:** O MDP é construído com a com as seguintes característica:

* Existe apenas uma possibilidade de ativo que é uma moeda estrangeira.
* O investidor é pequeno e não influencia o mercado com suas ações. (isso possibilita treinar o modelo sem investir dinheiro verdadeiro enquanto a estratégia ainda não funciona bem).
* O investidor sempre investe tudo.
* O investidor visa um horizonte de tempo infinito.

O MDP é descrito por um conjunto finito de estados e ações e um conjunto de probabilidades de transição. Cada estado possui uma Value-function V e o objetivo é encontrar a policy π que maximiza V para todos os estados.

O valor ótimo de V é calculado segundo duas abordagens:

1. Programação dinâmica, que assume que as probabilidades de transição e os valores esperados são conhecidos.
2. Q-learning, é um método de aprendizado por reforço que não exige um modelo do sistema conhecido. Ele busca a solução ótima amostrando pares de estados e retornos enquanto interage com o sistema.

Foram realizados dois experimentos distintos:

1. O primeiro experimento utiliza uma taxa de câmbio artificial. As probabilidades de transição foram definidas de modo que a taxa segue uma tendência crescente, mas quanto maior os valores, maior a chance de ocorrer uma queda brusca. Foram utilizados 2000 pontos de dados. Ambos os algoritmos foram testados nessa fase e apresentaram convergência com resultados satisfatórios. O algoritmo de programação dinâmica convergiu com apenas 5 iterações e apresentou solução idêntica ao QL.
2. O segundo experimento utilizou dados reais do German stock index DAX com a seguinte tarefa: investir em um bloco de ações que seguem o DAX ou investir no marco alemão. O DAX e mais 11 outros mercados influentes foram usados como inputs para o vetor de estados. Para armazenar as value functions do QL, foram usadas duas redes neurais, uma para cada ação, com 8 camadas ocultas e output linear. A título de comparação, foi utilizada uma estratégia de referência baseada em um modelo de Redes Neurais de 95, construído para prever mudanças diárias no DAX: se a previsão para o próximo dia é positiva, então o investimento é feito. A policy resultante do QL é claramente superior à estratégia de referência com 80% de retorno extra no fim do período de treino e 25% no período de teste.

**Contribuição e inovação:** Preciso ter mais clareza da linha do tempo para afirmar o quão pioneiro foi este trabalho, mas aparentemente foi um dos primeiros em aplicar q-learning combinado com redes neurais para gerenciamento de portfolios.

**Artigo:** A Multi-agent Q-learning Framework for Optimizing Stock Trading Systems   
**Autor:** Jae Won Lee and Jangmin O   
**Ano:** 2002

**Em geral:** Enquanto a maioria dos trabalhos de RL estudam apenas um agente, aqui temos uma aplicação para múltiplos agentes cooperativos, que se comunicam compartilhando training episodes e policies aprendidas, enquanto mantém o esquema convencional do QL. [inovação alert!] Os parâmetros de transação são otimizados com QL, utilizando RN aproximar os Q-values, uma vez que o número de pares de estados e ações é muito grande.

**Método:** Visa maximizar os ganhos dos investimentos considerando não apenas a tendência global mas também o movimento diário dos preços dos ativos. Isso é feito pelos múltiplos agentes, cada um com seu objetivo:

* Buy signal agent: faz predição, estimando informação de curto e longo prazo para produzir sinal de compra.
* Buy order agent: não faz predição, apenas determina um preço de compra.
* Sell signal agent: produz sinal de venda.
* Sell order agent: determina preço de venda.

Os estados são definidos como matrizes que representam os indicadores de longo e curto prazo. Quanto as ações, os agentes podem tomar dois tipos de ação: o Buy agent pode assumir BUY ou NOT-BUY, enquanto o sell agent pode HOLD ou SELL. Quando buy signal agent assume NOT-BUY, não recebe nenhuma recompensa. Quando assume BUY o cálculo da recompensa é adiado até o sell order agent vender o ativo. A recompensa é definida como a taxa de lucro, considerando os custos da transação.

**Experimento:** O método de QL é comparado como um método de aprendizado supervisionado de RN sendo ambos aplicados no mercado de ações Coreano. O experimento realizado com QL se mostrou superior em retorno e gerenciamento de risco, mesmo em um momento de queda brusca do mercado, enquanto o método RN perde severamente, o QL apresenta perdas pequenas.

**Inovação:** Lee e Jangmin inovam trazendo uma aplicação de deep q-learning para múltiplos agentes.

**Artigo:**  An Algorithm for Trading and Portfolio Management Using Q-learning and Sharpe Ratio Maximization  
**Autor:** Xio Gao, Laiwan Chan  
**Ano:** 2000

**Em geral:** O trabalho propõe um método (QRS) de gerenciamento de portfólio utilizando QL e maximização de Sharpe Ratio.

Alguns estudos como o de Neuneir98 utiliza QL para maximizar o lucro, porém assume que o investidor não tem aversão ao risco, o que não funciona bem em ambientes com alto risco. Aqui os autores propõem uma forma de mensurar o risco utilizando uma medida de lucro relativo ajustado ao risco. [inovação alert!] O Sharpe Ratio é uma forma comum dessa medida e é dado pela razão do retorno médio pelo desvio padrão do retorno.

Embora o Q-learning seja um método não supervisionado, a maximização do sharpe ratio é feita com aprendizado supervisionado usando um input "dados desejados".