# Opdracht 1: CNN Parameters

A diagram of a diagram

Description automatically generatedA screenshot of a computer code

Description automatically generateda.

b. De 1e laag bestaat uit 32 filters om features te herkennen in de afbeelding. Daarna wordt max pooling toegepast om het resultaat te comprimeren. De laag erna bestaat uit 64 filters om opnieuw features te bepalen aan de hand van de huidige tensor. De 4e laag is nog een pooling laag om de data verder te comprimeren. Om de data als input te gebruiken in het “fully connected” neuraal netwerk, wordt de data omgezet naar een vector door de flatten laag. Vervolgens is er één hidden laag (128 neuronen) die patronen gaat herkennen in de data en tot slot is er een output laag van 10 neuronen.

c.

Convolutie:  
1 filter/kernel bestaat uit 9 cijfers (3 x 3). 32 x 9 = 288 parameters in de 1e laag. Pooling laag heeft geen parameters (wordt alleen gebruikt voor comprimeren). Vervolgens: 64 x 9 = 576 voor de 3e laag. De pooling laag en flatten laag erna hebben geen trainbare parameters.

Flatten laag:  
Om het aantal weights en biases te bepalen, moet ik weten hoe groot de vector is na de flatten laag (dit is namelijk de input voor het fully connected neuraal netwerk):

Afbeelding start als (28, 28, 1). Uitgaande van een stride van 1 worden de dimensies na de eerste convolutie laag (26, 26, 32). De randen vallen weg en er zijn 32 filters die door de data gaan. Pooling comprimeert dit tot (13, 13, 32). De 2e convolutie laag maakt hier (11, 11, 64). Randen vallen weer weg en er zijn 64 filters. De laatste pooling laag comprimeert dit tot (5, 5, 64). Tot slot maakt de flatten laag van deze data een 5 x 5 x 64 = 1600 lange vector!

Fully connected netwerk:  
Voor de eerste hidden laag zijn er 1600 x 128 = 204800 weights en 128 biases, 204928 parameters dus. Voor de laatste laatste laag zijn er 128 x 10 = 1280 weights en 10 biases, 1290 parameters dus.

**Totaal**:  
288 + 576 + 204928 + 1290 = 207.082 trainbare parameters in totaal!

# Opdracht 2: Hyper parameter optimization with GA

Voor een neuraal netwerk is het belangrijk om te bepalen wat de beste hyperparameters zijn om het model te trainen. Dit zijn waardes die voorafgaand aan het trainen vast staan en tijdens het trainen niet meer veranderen. Een effectieve methode om deze waardes te bepalen is om een “Genetic Algoritm” (GA) toe te passen.

Een GA houdt in dat er een bepaald aantal “Chromosomen” of “Organisme” tot leven worden gewekt. Deze bestaat uit een aantal “Genen” die willekeurig gezet worden. Van elk organisme wordt geëvalueerd hoe goed deze presteert. Vervolgens worden met de beste (in het geval van seksuele voortplanting de beste 2) organisme gebruikt om nieuwe organisme te maken die de genen van hun “ouders” overnemen, met een bepaalde kans op een mutatie.

In het geval van het tunen van hyperparameters voor een neuraal netwerk, kan elke hyperparameter gezien worden als 1 “gen” van het organisme. Denk bij hyperparameters aan instellingen zoals het aantal convolutielagen, het aantal dense lagen en de learning rate. Het doel is om het best presterende organisme te vinden, want deze bevat de meest geschikte parameters voor het trainen van het neuraal netwerk.

Hoewel organismes beginnen met willekeurige genen, is het belangrijk dat er restricties zijn voor welke waardes deze kunnen hebben. Bij een CNN is het bijvoorbeeld noodzakelijk dat er minstens 1 convolutie laag is, deze parameter mag dus nooit op 0 gezet worden bijvoorbeeld.

Met deze restricties bepaald, kunnen de organismes gemaakt worden. Van elk organisme wordt de “fitness” bepaald. In deze context gebeurd dat door het neuraal netwerk te laten trainen met de genen van het organisme als parameters en het resultaat hiervan te evalueren. De organismes met de beste fitness worden gebruikt om nieuwe organisme te maken. Deze hebben allemaal een kans op een mutatie, wat deze mutatie kan zijn wordt vooraf afgesproken. De kans op een mutatie is afhankelijk van de mutatie rate die is afgesproken.

Zodra we tevreden zijn over de fitness van een organisme, kan het GA stopgezet worden. De genen van dit organisme kunnen vervolgens gebruikt worden als hyperparameters om een model te trainen, dit zou moeten lijden tot een zo goed mogelijk model.