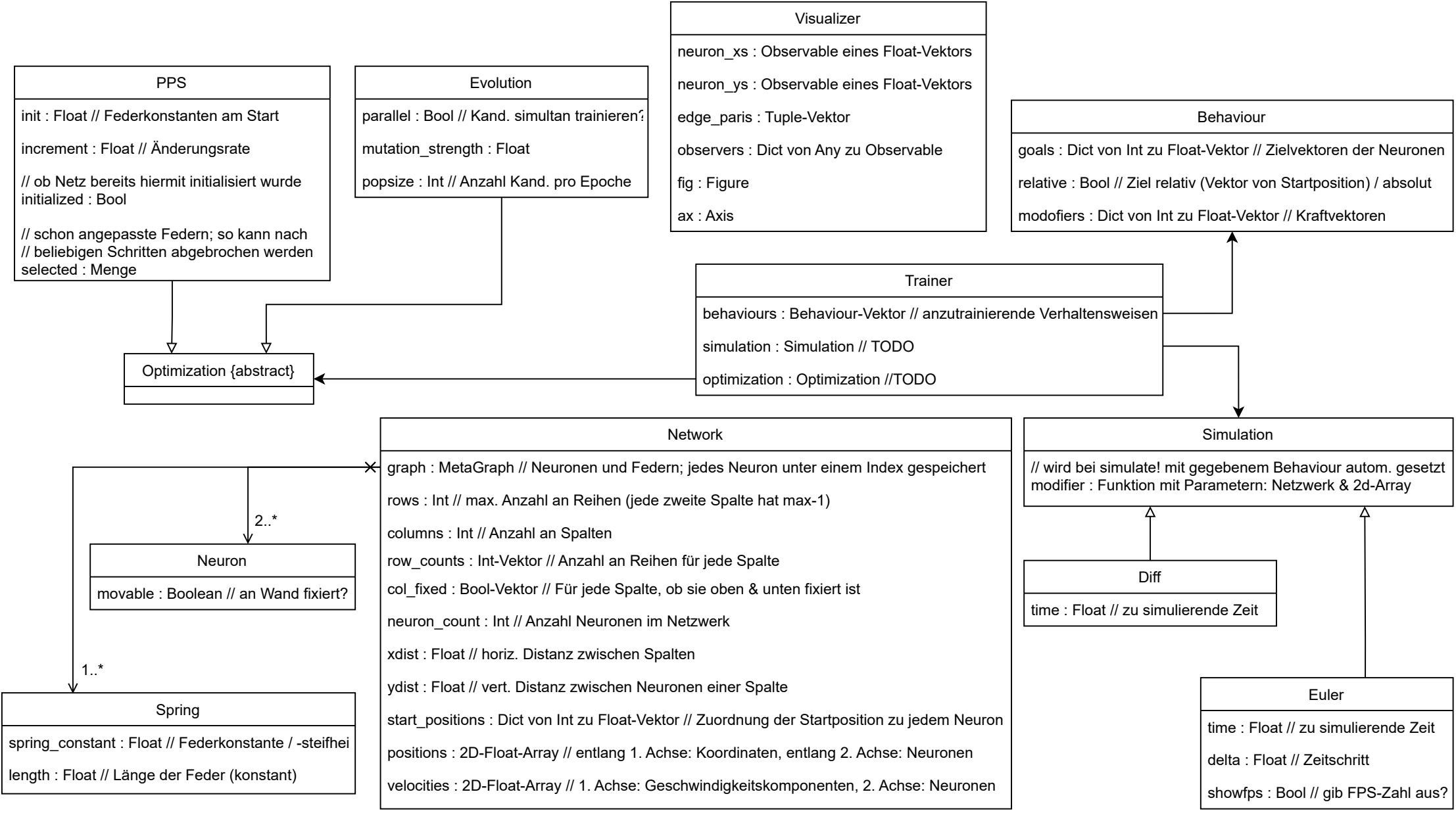
Visualizer
neuron_xs : Observable eines Float-Vektors
neuron_ys : Observable eines Float-Vektors
edge_paris : Tuple-Vektor
observers : Dict von Any zu Observable
fig : Figure
ax : Axis
PPS
init : Float // Federkonstanten am Start
increment : Float // Änderungsrate
// ob Netz bereits hiermit initialisiert wurde
initialized : Bool
// schon angepasste Federn; so kann nach
// beliebigen Schritten abgebrochen werden
selected : Menge
Evolution
parallel : Bool // Kand. simultan trainieren?
mutation_strength : Float
popsize : Int // Anzahl Kand. pro Epoche
Behaviour
goals : Dict von Int zu Float-Vektor // Zielvektoren der Neuronen
relative : Bool // Ziel relativ (Vektor von Startposition) / absolut
modofiers : Dict von Int zu Float-Vektor // Kraftvektoren
Trainer
behaviours : Behaviour-Vektor // anzutrainierende Verhaltensweisen
simulation : Simulation // TODO
optimization : Optimization //TODO
Optimization {abstract}
Network
graph : MetaGraph // Neuronen und Federn; jedes Neuron unter einem Index gespeichert
rows : Int // max. Anzahl an Reihen (jede zweite Spalte hat max-1)
columns : Int // Anzahl an Spalten
row_counts : Int-Vektor // Anzahl an Reihen für jede Spalte
col_fixed : Bool-Vektor // Für jede Spalte, ob sie oben & unten fixiert ist
neuron_count : Int // Anzahl Neuronen im Netzwerk
xdist : Float // horiz. Distanz zwischen Spalten
ydist : Float // vert. Distanz zwischen Neuronen einer Spalte
start_positions : Dict von Int zu Float-Vektor // Zuordnung der Startposition zu jedem Neuron
positions : 2D-Float-Array // entlang 1. Achse: Koordinaten, entlang 2. Achse: Neuronen
velocities : 2D-Float-Array // 1. Achse: Geschwindigkeitskomponenten, 2. Achse: Neuronen
Simulation
// wird bei simulate! mit gegebenem Behaviour autom. gesetzt
modifier : Funktion mit Parametern: Netzwerk & 2d-Array
2..*
Neuron
movable : Boolean // an Wand fixiert?
1..*
Spring
spring_constant : Float // Federkonstante / -steifhei
length : Float // Länge der Feder (konstant)
Diff
time : Float // zu simulierende Zeit
Euler
time : Float // zu simulierende Zeit
delta : Float // Zeitschritt
showfps : Bool // gib FPS-Zahl aus?