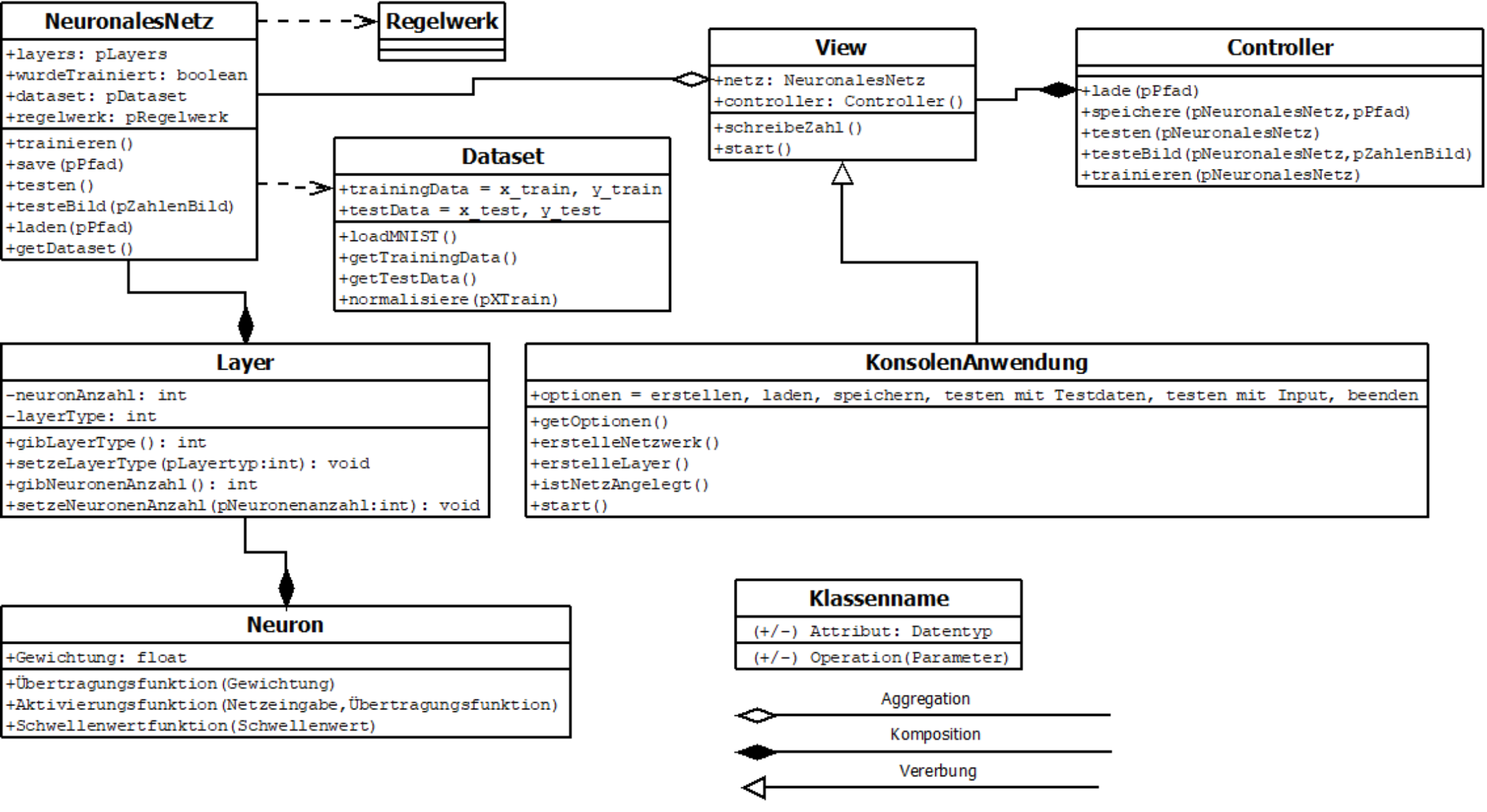
NeuronalesNetz
+layers: pLayers
+wurdeTrainiert: boolean
+dataset: pDataset
+regelwerk: pRegelwerk
+trainieren()
+save(pPfad)
+testen()
+testeBild(pZahlenBild)
+laden(pPfad)
+getDataset()
Regelwerk
View
+netz: NeuronalesNetz
+controller: Controller()
+schreibeZahl()
+start()
Controller
+lade(pPfad)
+speichere(pNeuronalesNetz,pPfad)
+testen(pNeuronalesNetz)
+testeBild(pNeuronalesNetz,pZahlenBild)
+trainieren(pNeuronalesNetz)
Dataset
+trainingData = x_train, y_train
+testData = x_test, y_test
+loadMNIST()
+getTrainingData()
+getTestData()
+normalisiere(pXTrain)
Layer
-neuronAnzahl: int
-layerType: int
+gibLayerType(): int
+setzeLayerType(pLayertyp:int): void
+gibNeuronenAnzahl(): int
+setzeNeuronenAnzahl(pNeuronenanzahl:int): void
KonsolenAnwendung
+optionen = erstellen, laden, speichern, testen mit Testdaten, testen mit Input, beenden
+getOptionen()
+erstelleNetzwerk()
+erstelleLayer()
+istNetzAngelegt()
+start()
Neuron
+Gewichtung: float
+Übertragungsfunktion(Gewichtung)
+Aktivierungsfunktion(Netzeingabe,Übertragungsfunktion)
+Schwellenwertfunktion(Schwellenwert)
Klassenname
(+/-) Attribut: Datentyp
(+/-) Operation(Parameter)
Aggregation
Komposition
Vererbung