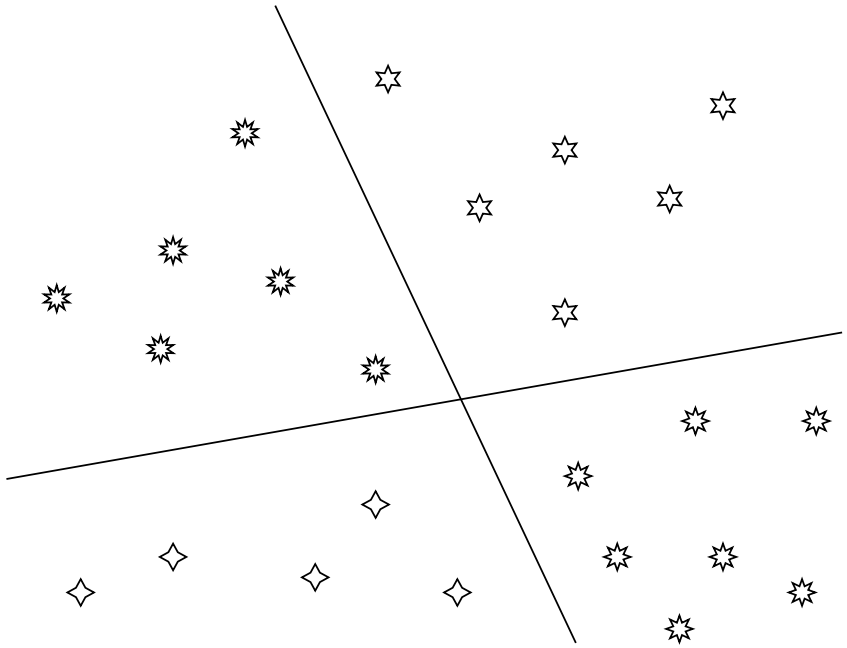
Perceptrón

- *Entradas preprocesadas por unidades de asociación (llamadas "unidades A").*
- *Patrones binarias de entrada.*
- *Las unidades A realizan operaciones binarias (lógicas) fijas cualesquieras.*



Separación a más clases

- *Utilizamos más nodos en la red.*
  - *Cada nodo separa una clase de los demás.*
- *Si cada una de estas las clases es linealmente separable de los demás, basta con TLUs/perceptrones.*
- *También sirven cuando las combinaciones de los híperplanos logran "aislar" las clases.*
  - *Se va a ocupar más capas para separar las combinaciones.*
    - *La capa de salida puede ser fija o entrenada.*

Procesamiento de imágenes

- *Reducción de colores.*
- *Filtrado para reducir ruido.*
- *Reducción de resolución.*
- *Representación binaria.*

Método de gradiente

- *Objetivo: min* $\mathbf{y} = f(\mathbf{x})$.
- *Denotamos el óptimo por* $\mathbf{x}_0$.
- *Denotamos un candidato conocido por* $\mathbf{x}^*$.
- *Un cambio* $\Delta\mathbf{x}$ *aplicado a* $\mathbf{x}^*$*; examinamos su efecto* $\Delta\mathbf{y}$ *comparando* $f(\mathbf{x}^*)$ *con* $f(\mathbf{x}^* \pm \Delta\mathbf{x})$.
- *Aceptamos aquel cambio que proporciona mejora.*
- *La mejora proviene de la dirección del tangente:*
  - $\Delta\mathbf{x} = -\alpha f'(\mathbf{x}^*) \Rightarrow \Delta\mathbf{y} \approx f(\mathbf{x}^*)\,\Delta\mathbf{x} = -\alpha (f'(\mathbf{x}^*))^2$, *donde* $f'(\mathbf{x}^*)$ *es un véctor de las derivadas parciales* $\partial y/\partial x_i$ *evaluadas en* $\mathbf{x}^*$.

Minimización del error

- *Aplicamos el método de gradiente en una función que mide el error de clasificación de una red neuronal.*
  - $\Delta w_i = -\alpha\,(\partial E / \partial w_i)$, *donde* $E = \frac{1}{N}\sum_{p=1}^{N} e^p$.
    - $e^p = (t^p - a^p)^2/2$, *por ejemplo;*
      - *aquí la activación a sigue la función sigmoidal desde -1 hasta 1 para proveer un error suave.*
- *Esto es computacionalmente demandante.*

Regla delta

- *Para poder entrenar un patrón a la vez.*
- *Usamos para nodo j como un estimado del gradiente solamente*

$$\frac{\partial e^p}{\partial w_{ji}} = -(t^p - a^p)x^p_{ji},$$

  - *donde* $x_i{}^p$ *es el componente i del patrón p.*
- *Esto nos da*

$$\Delta w_{ji} = \alpha(\sigma'(a^p_j))(t^p_j - y^p_j)x^p_{ji}.$$

Entrenamiento de dos capas

- *Usando dos conjuntos de patrones de entrenamiento, se entrenan dos capas; cada capa tiene su propio entrenamiento.*
  - *Se aplica el método de gradiente.*
  - *La complicación es la evaluación del gradiente para los nodos "ocultos" de la capa interior.*
  - *Podemos usar, con una definición adecuada de* δ, *la siguiente formulación:*

$$\Delta w_{ki} = \alpha(\sigma'(a^p_k))\delta^p_k x^p_{ki}.$$

Problema de asignación de crédito

- *Errores qué tan graves en cuáles nodos del conjunto I^k que toma entradas del nodo oculto k fueron producidos por una mala decisión de su parte:*

$$\delta_k^p = \sigma'(a_k^p) \sum_{j \in I_k} \delta_j^p w_{jk}.$$

Errores de la capa de salida

$$\delta_k^p = \sigma'(a_k^p)(t_k^p - y_k^p)$$

Algoritmo

- *Asigne valores iniciales pequeños uniformemente al azar.*
  - *Tan pequeñas que ningún patrón de entrenamiento provoca salidas cercanas a los valores extremos.*
- *Realice los "epochs", es decir, presente patrones uno por uno hasta que el error llegue a un nivel aceptable.*
- *Paso adelante:*
  - *Evalúe los nodos ocultos.*
  - *Evalúe los nodos de salida.*
- *Compara la salida con la meta.*
- *Paso atrás:*
  - *Calcula los errores de las salidas.*
  - *Calcula los errores de los nodos ocultos.*
  - *Aplica el método de gradiente.*

Mínimos locales vs. globales

- *El método de gradiente es un algoritmo local.*
  - *Podemos aplicar reinicios.*
  - *La aplicación de metaheurísticos para escapar óptimos locales.*
- *También el criterio de paro requiere consideración.*
  - *Error máximo / promedio / estancamiento...*

Adaptación

- *Aprendizaje de momentum:*
  $\Delta w_{ji}(n) = \alpha \delta_j^p x_{ji}(n) + \lambda \Delta w_{ji}(n-1),$
  - *donde n es el epoch (iterativo).*
    - *"Agarra momentum" cuando los cambios van en la misma dirección, lo pierde si van en direcciones opuestas.*

Variantes

- *Más capas.*
- *Variaciones en la conectividad entre capas.*
  - *Incompletos.*
  - *Brincando capas.*
- *Interpretando los nodos ocultos como "extractores de caracterícticas".*

Sobreajuste

- *Todos los datos de prueba están correctamente clasificados, pero las otras muestras fallan.*
  - *Debido a que se ha ajustado en mucho detalle para acomodar hasta el ruido por tener un exceso de grados de libertad.*
    - *Hay que evitar un exceso en los nodos internos.*

Decisiones a tomar

- *¿Cuántas capas?*
- *¿Cuántos nodos por capa?*
- *¿Cómo se conectan?*
- *¿Cómo se entrenan?*
- *¿Con cuántas muestras / cuándo se termina el entrenamiento?*