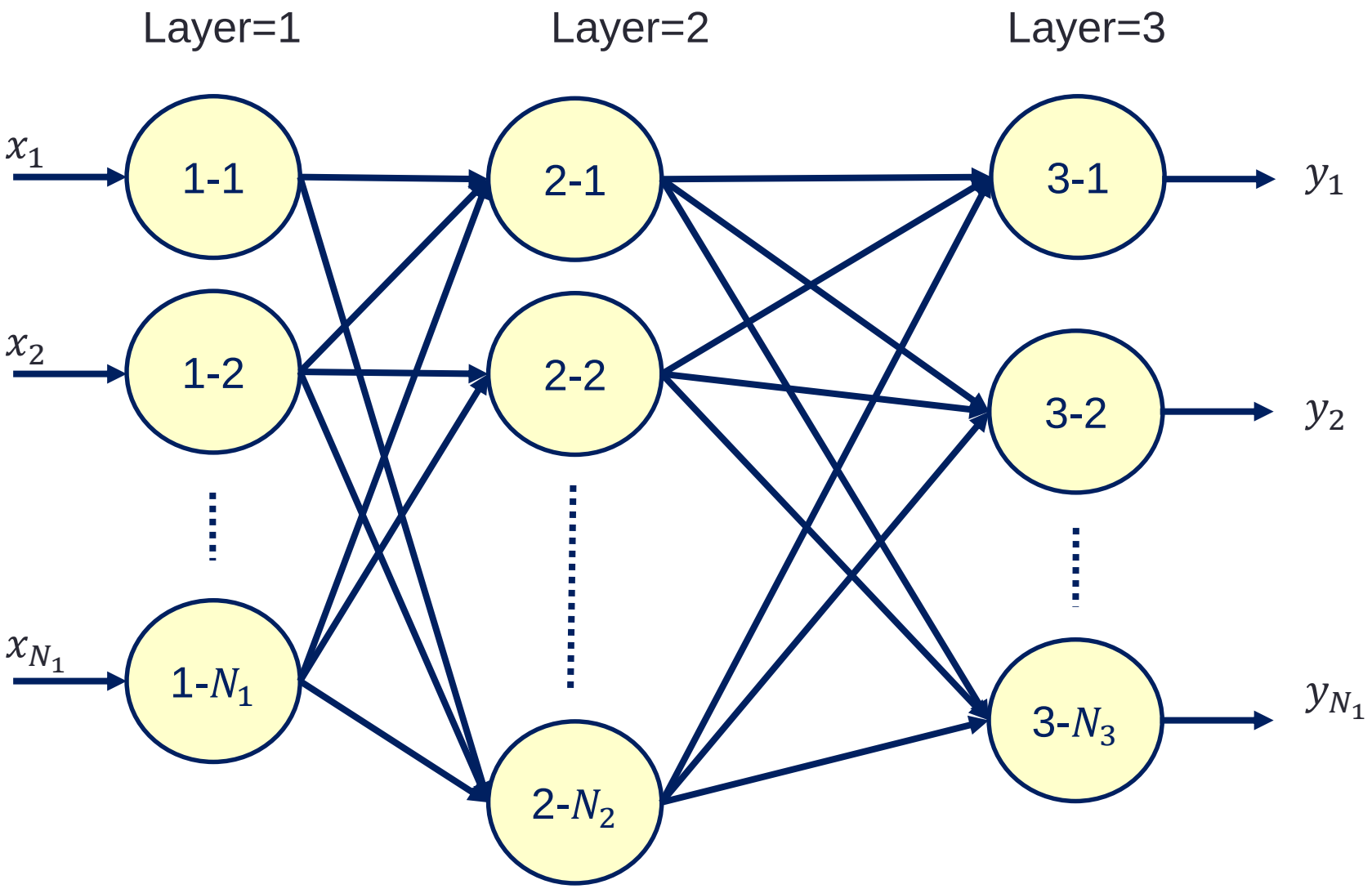
Neural Network and GEANT4

河野能知

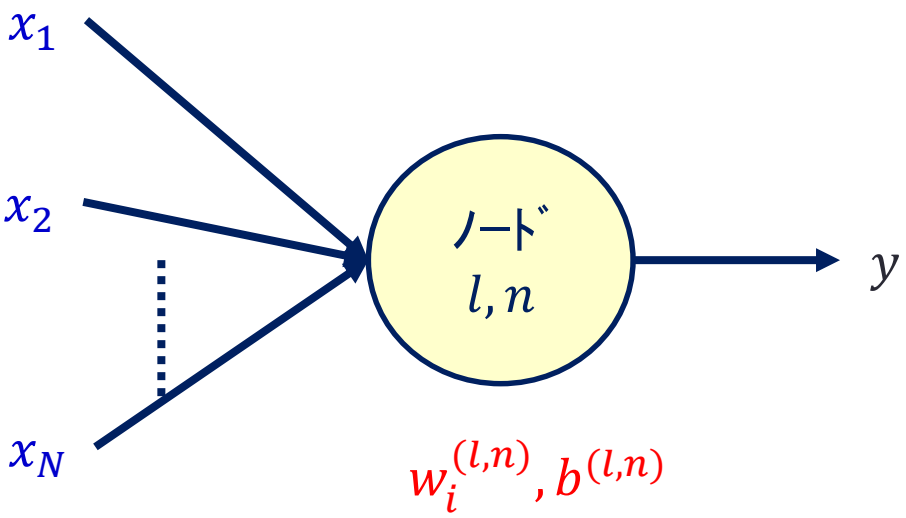
2016.12.02

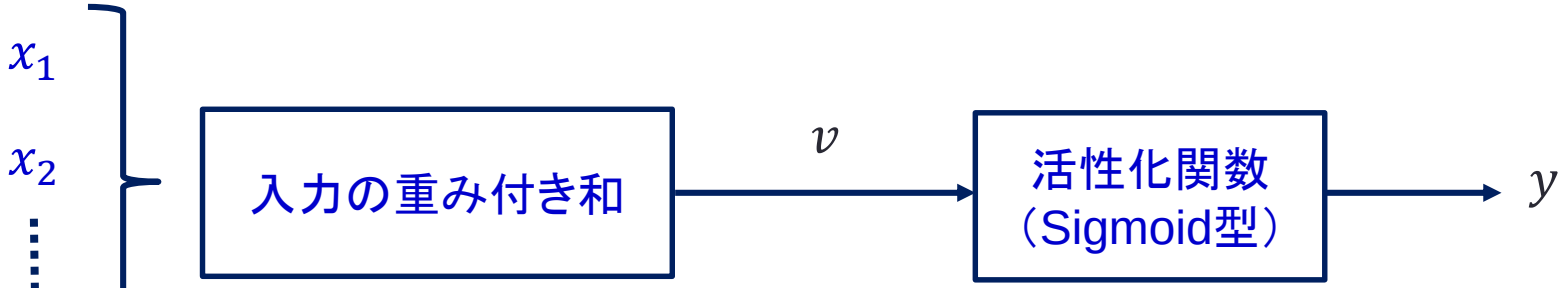
研究室ミーティング

Neural network



各ノードで行う計算



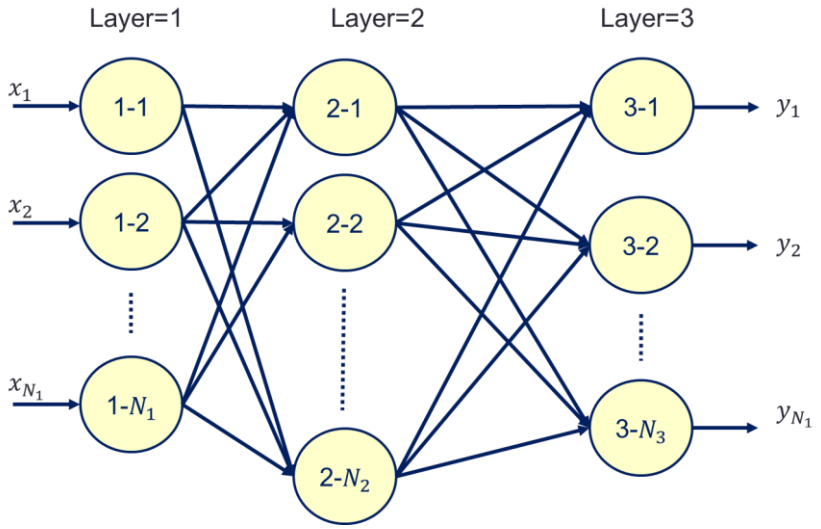


$$v = \sum_{i=1}^{N} w_i^{(l,n)} x_i - b^{(l,n)}$$

$$y = \frac{1}{1 + e^{-v}}$$

ニューラルネットワークの目的

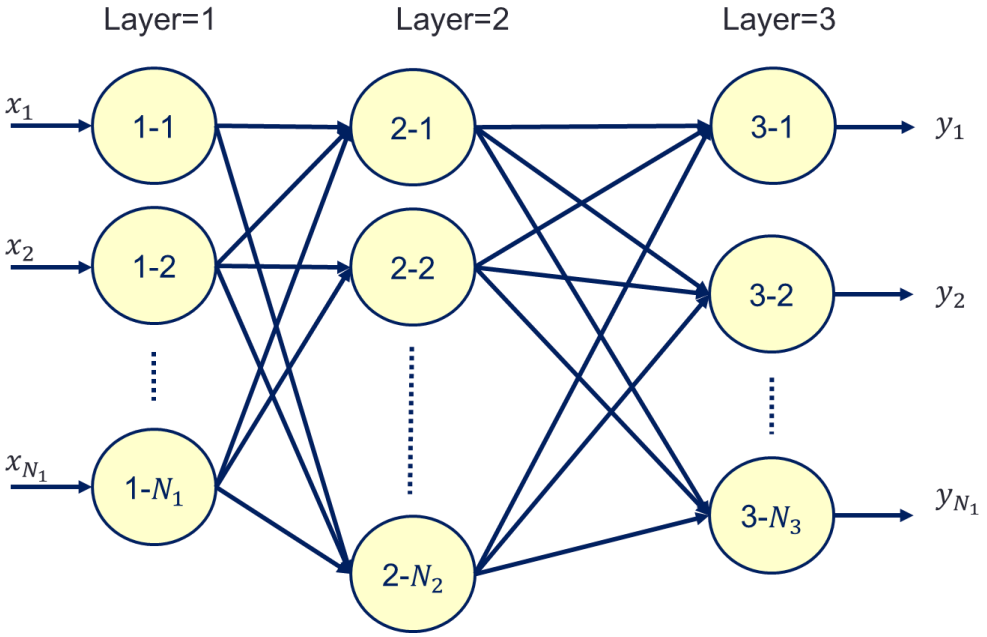
- ニューラルネットワークの構成
  - 入力層（入力変数の数だけ）
  - 出力層（1個以上）
  - いくつかの中間層
- 各ノードの重みと閾値の値によって、入力から様々な出力を得ることができる
- 比較的簡単な計算によって非線形な関数を表現することができう
- 中間層を増やすことによって、任意の関数形を表現できる



- 出力が1個だけのものは、類別問題に適用されることが多い
  - クラスA → 出力=0
  - クラスB → 出力=1

ニューラルネットワークの訓練

- 入力データに対して出力値が分かっているサンプルを用いて、望みの出力が得られるように重みを最適化する
  - → 訓練（トレーニング）



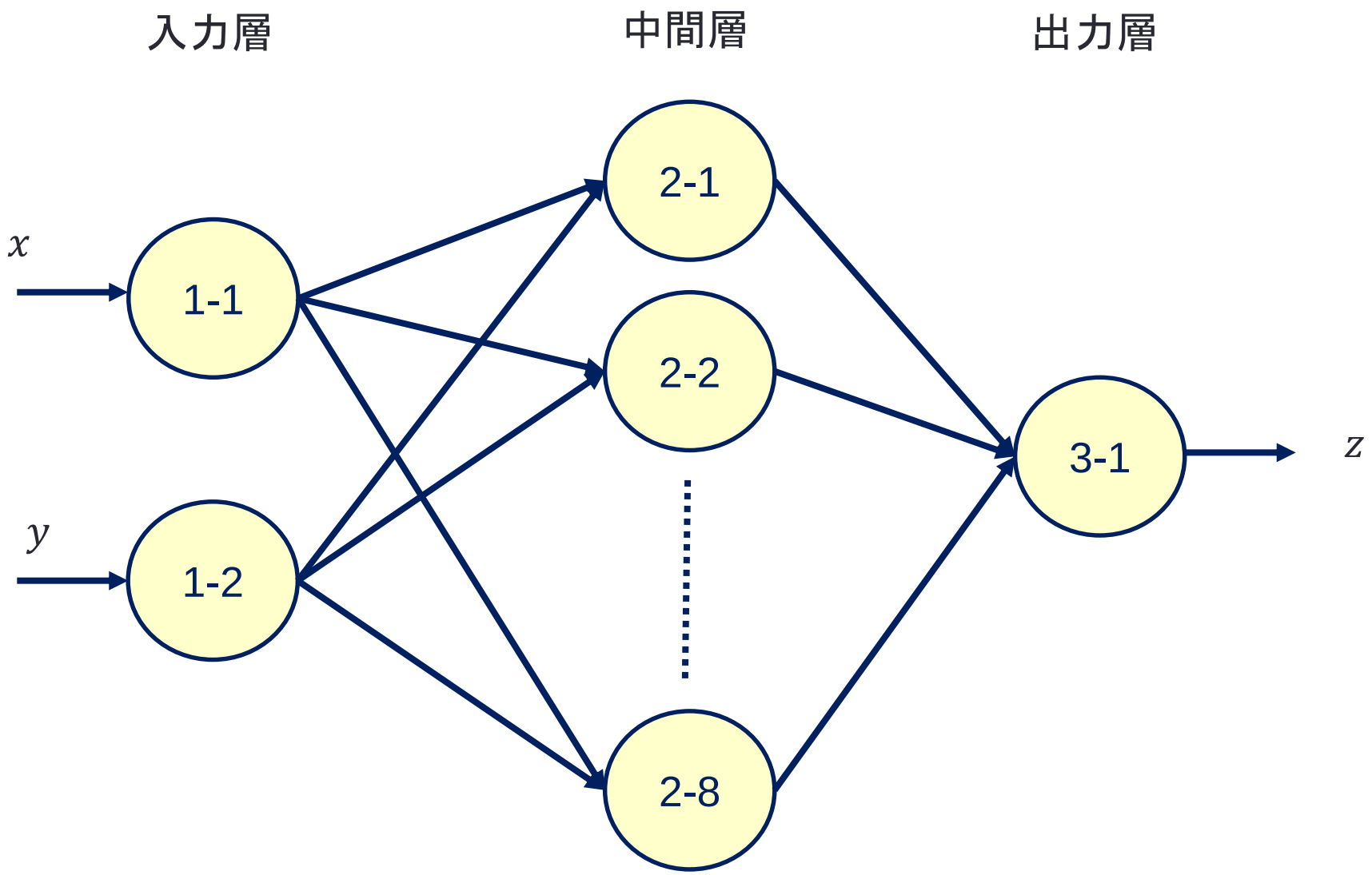
$$\left(x_1^{(1)}, x_2^{(1)}, \ldots;\ y^{(1)}\right)$$
$$\left(x_1^{(2)}, x_2^{(2)}, \ldots;\ y^{(2)}\right)$$
$$\left(x_1^{(3)}, x_3^{(3)}, \ldots;\ y^{(3)}\right)$$
$$\ldots\ldots$$

$$\left\{w_i^{(l,n)}, b^{(l,n)}\right\}$$

残差＝正しい出力値 -- 入力から計算した出力値
を最小化するように各ノードのパラメータを決める

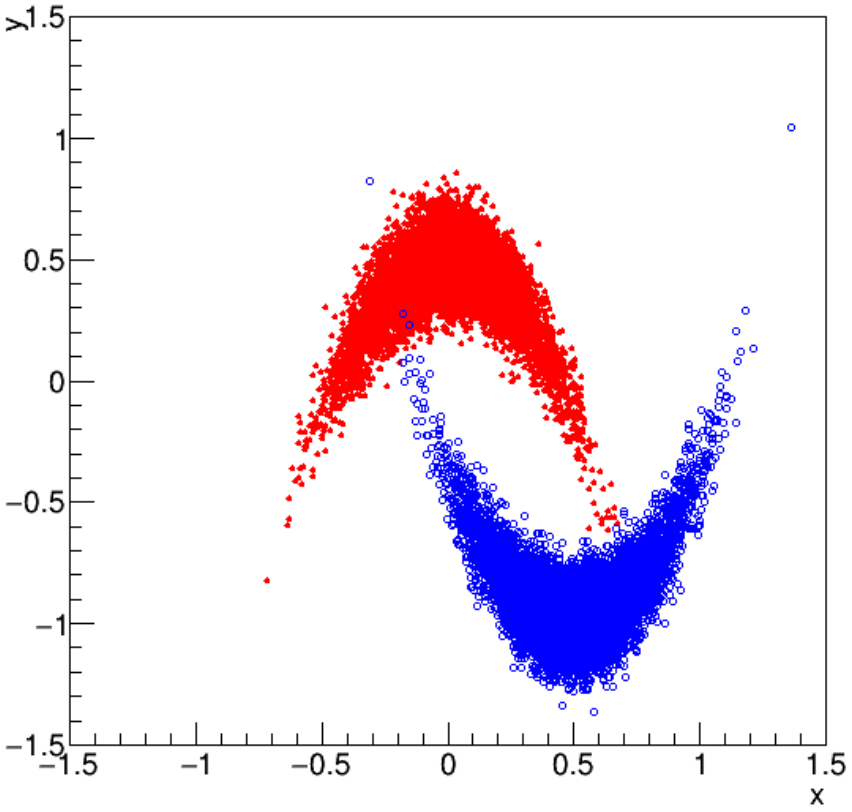
誤差逆伝播法

2入力→中間層(8ノード)→1出力

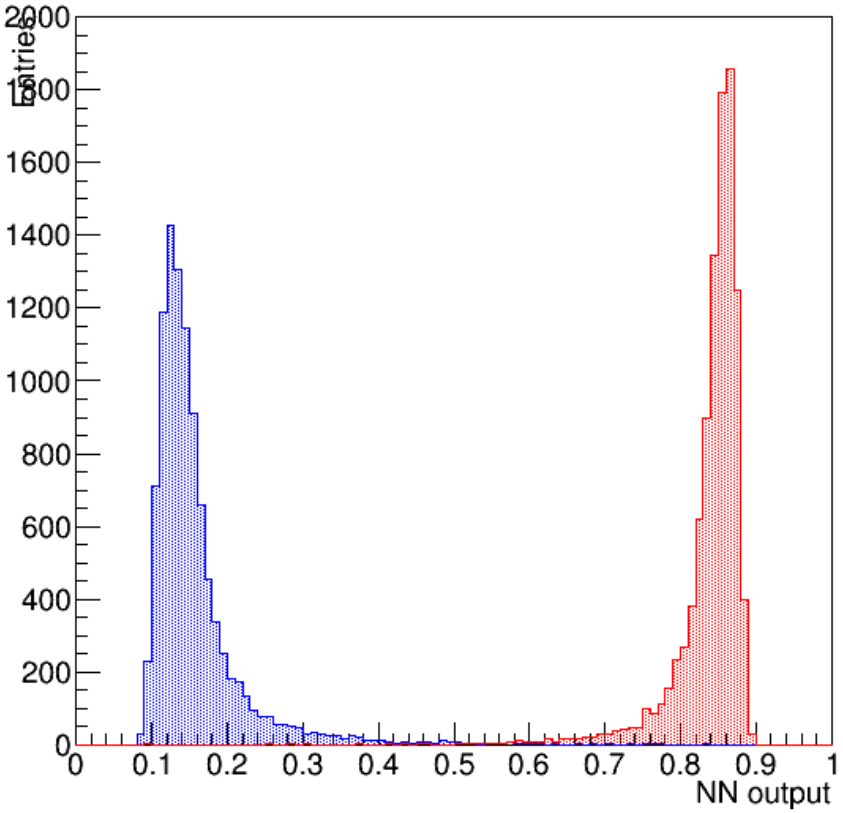


2次元分布の類別

2次元の座標が与えられたときに、赤丸と青丸のような2つのサンプルのどちらに属するかを判定したい



赤丸のサンプル ➔ 出力1
青丸のサンプル ➔ 出力0
となるようにトレーニングした後のNNの出力



飛跡再構成法への応用

- 飛跡の再構成
  - 検出器のヒット情報（位置情報）から、荷電粒子の飛跡を再構成する
- 例
  - 複数のヒットの座標が与えられたときに、それらが同一飛跡に属するかどうかを判定する
  - 飛跡パラメータと新たなヒットの座標から、より正確な飛跡パラメータを推定する

- 参考
  - 2010年頃から、飛跡の再構成において新たなアルゴリズムやFPGAやGPUを利用した並列化が議論されている
  - https://ctdwit2017.lal.in2p3.fr/
  - それに特化した会議： Connecting the Dots, Intelligent Trackers

GEANT4

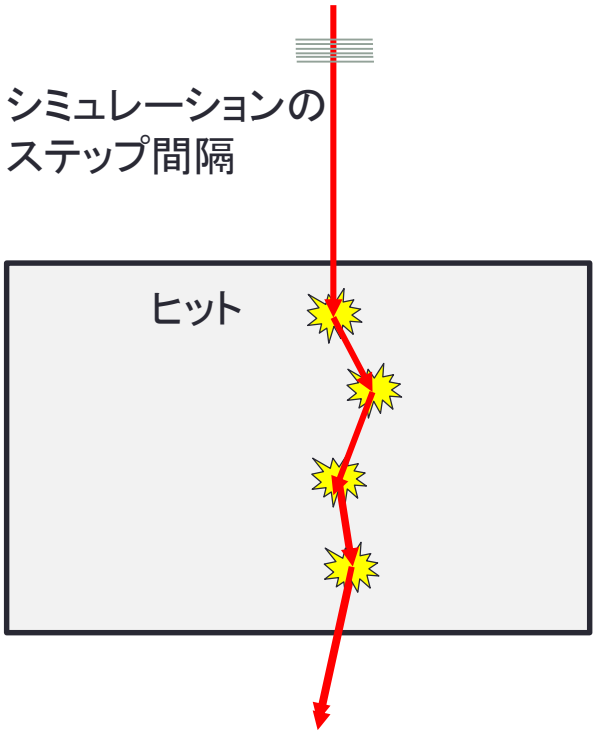
ユーザーが行うこと: 次の機能をクラスのメンバ関数として定義する

1. 検出器をセットアップする
2. シミュレーションに使う粒子と相互作用を指定する
3. 検出器に入射する粒子を生成する

GEANT4が行うこと: main関数の中でユーザー定義の関数を呼び出す

1. ユーザーが定義した関数を呼び出して、プログラム内で検出器のモデルを作る
2. 指定された事象数分だけ、粒子の生成と検出器との相互作用のシミュレーションを繰り返す
    1. 粒子の生成にはユーザー関数を呼び出す
    2. 指定された相互作用のみを扱う

粒子と物質との相互作用のシミュレーション



- 相互作用の断面積をもとに、反応の起こり易さを見積もる
- 粒子を運動量方向に少しずつ進めて行く(step)
- 物質中をある厚みd進む間に反応の起こる確率が分かっているので、その確率で反応を起こしたり起こさなかったりする
- 反応を起こした場合には、Hitを作る
  - 物質にエネルギー損失がある
  - そこで粒子の運動量が変わる
- シミュレーションの結果、大量のG4VHitが生成され、そこからいろいろと知ることができる
  - 各物質でのエネルギー損失
  - ヒット前後の粒子のエネルギー・運動量

Sensitive Detector

- 測定器全体の中で、測定を行う上での有感領域に相当する
  - センサー以外の部分で起こった反応を全て調べる必要はない
  - フレームやケーブルで起こった反応は、エネルギー損失等は計算するけど、そこで起こった反応の詳細（G4VHit）を全て知る必要はない
- 測定結果と比較するためには、有感領域で生成されたヒットのみを保存すればよい
  - 検出器をSensitiveDetectorに指定することができる
  - そして、その中で生成したヒットのみを保存する

G4UserRunAction, G4UserEventAction

- G4UserEventAction
  - void BeginOfEventAction(const G4Event* event);
  - void EndOfEventAction(const G4Event* event);
  - の2つの関数を子クラスで実装する。そうすると、事象の最初（粒子を生成する前）と最後（相互作用を全てシミュレートした後）でこれらの関数が呼び出される
  - EndOfEventAction(…)の中で結果をhistogramやTTreeに保存する