

Katalóg požiadaviek

Obsah:

- 1 Úvod
 - 1.1 Účel tohto katalógu požiadaviek
 - 1.2 Rozsah využitia systému
 - 1.3 Slovník pojmov
 - 1.4 Odkazy a referencie
 - 1.5 Prehľad nasledujúcich kapitol
- 2 Všeobecný popis
 - 2.1 Perspektíva systému
 - 2.2 Funkcie systému
 - 2.3 Charakteristika používateľa
 - 2.4 Všeobecné obmedzenia
 - 2.5 Predpoklady a závislosti
- 3 Špecifické požiadavky
 - 3.1 Funkčné požiadavky
 - 3.2 Kvalitatívne požiadavky
 - 3.3 Požiadavky rozhrania
- 4 Prílohy

Katalóg požiadaviek

Úvod

1.1 Účel tohto katalógu požiadaviek

Tento dokument slúži ako katalóg požiadaviek pre aplikáciu Rekonštrukcia Top Kvarkov, ktorá slúži ako projekt pre predmet Tvorba informačných systémov.

1.2 Rozsah využitia systému

Cieľom produktu je vytvorenie systému, ktorý bude pomocou neurónových sietí odhadovať, ktoré jety vznikli z určitých top kvarkov. Ďalším cieľom je porovnanie výstupu z KLFitter-u a neurónových sietí v webovom rozhraní, kde sa zabezpečí plynulosť aplikácie. Webové rozhranie bude pracovať s databázou, do ktorej bude ukladať výsledky z naprogramovaných algoritmov.

1.3 Slovník pojmov

Kvark - podľa štandardného modelu časticovej fyziky elementárne častice, z ktorých sa skladajú hadróny (teda napríklad protóny a neutróny).

Top kvark – je kvark tretej generácie s elektrickým nábojom $+(2/3)e$. Je to najťažší zo všetkých známych elementárnych častíc.

Jet - úzky kužeľ hadrónov a iných častíc produkovaných hadronizáciou kvarku alebo gluónu v experimente s fyzikou častíc alebo experimentom s ťažkými iónmi.

KLFitter – knižnica na kinematickú montáž pomocou pravdepodobnosti. Je primárne vyvinutý pre prípad rekonštrukcie top kvarku, ale dá sa ľahko upraviť tak, aby vyhovoval aj iným procesom.

Neurónová sieť – výpočtový model, zostavený na základe abstrakcie vlastností biologických nervových systémov.

Tensorflow – bezplatná a otvorená softvérová knižnica pre dataflow a diferencovateľné programovanie v rôznych úlohách. Je to symbolická matematická knižnica a používa sa tiež na aplikácie strojového učenia, ako sú neurónové siete.

KERAS – vysokoúrovňové API pre neurónové siete, schopné bežať na vrchole Tensorflow. Umožňuje rýchle experimentovanie prostredníctvom vysokoúrovňového, ľahko použiteľného, modulárneho a rozšíriteľného API.

1.4 Odkazy a referencie

1. **ROOT**: <https://root.cern.ch/>
2. <https://root.cern.ch/how/how-read-tree>
3. **KLFitter**: <https://github.com/KLFitter/KLFitter>
4. **LWTNN**: <https://github.com/lwttn/lwttn>

Katalóg požiadaviek

1.5 Prehľad nasledujúcich kapitol

V nasledujúcich kapitolách sa čitateľ oboznámi s funkciami , perspektívami a obmedzeniami systému, taktiež so špecifickými požiadavkami.

Katalóg požiadaviek

Všeobecný popis

Optimalizácia neurónovej a metódy KL Fitter siete slúžiacej na rekonštrukciu top kvarkových párov.

2.1 Perspektíva systému

V tomto systéme bude mať užívateľ 2 možnosti výberu rekonštrukcie: 1. cez dnn

2. cez KL Fitter

Pospracovaní sa výsledky uložia do databázy a výstupom bude tabuľka na kontrolu ako dobre boli dané jety sparované s kvarkami

2.2 Funkcie systému

V tomto systéme bude mať užívateľ rozhranie, v ktorom bude mať možnosť vybrať si akú sieť chce trénovať a ďalšie možnosti ako napr. premenné, počet vrstiev atď.. Výstupom rozhrania bude čitateľný textový súbor aby bola možnosť skontrolovať proces aj bez webového rozhrania.

2.3 Charakteristika používateľa

Tento systém je určený pre fyzikálnych výskumníkov a študentov, ktorí pracujú s top kvarkami a jetmi.

2.4 Všeobecné obmedzenia

Tento systém potrebuje server, databázu a pripojenie na internet.

2.5 Predpoklady a závislosti

1. úložný priestor na serveri
2. internet
3. server na ktorom bude bežať program

Katalóg požiadaviek

Špecifické požiadavky

3.1 Funkčné požiadavky:

KIFitter: Implementácia KIFitteru, ktorá bude robiť rekonštrukciu top kvarkov.

DNN: Implementácia neuronových sietí sa bude vedieť natrénovať, a robiť rekonštrukciu top kvarkov, z výsledku by sa malo dať porovnať, ako dobre to neurónové siete zvládli. Teda porovnať zo z výsledkom z KLFitter.

Databáza: Výsledky z algoritmov budú uložené v database, taktiež sa tam bude ukladať ako dlho process bežal.

Spúšťanie: Samotné spustenie algoritmu by sa malo dať aj cez linux(command line)

Premenné: Vo web interface sa budú dať zvoliť alebo zmeniť premenné použité v algoritmoch

Human readable: Musíme zabezpečiť, aby bol systém human readable, a teda aby sme vedeli výsledky zhodnotiť aj ručne.

Vstup: Vstup budeme musieť upraviť pomocou už implementovanej knižnice (vstup je vo forme ROOT), aby sme s ním vedeli pracovať vo zvolenom jazyku. A teda aby s dátami vedeli pracovať KERAS.

Trénovanie: Výstup z tréovania DNN treba prekonvertovať aby bol použiteľný v C++ (LWTNN [4])

3.2 Kvalitatívne požiadavky:

Procesy: Systém by mal vedieť spustiť viac procesov (Vstupov) naraz.

Stabilita: Systém by nemal padať.

Rýchlosť: Systém by mal pracovať čo najefektívnejšie, pre veľké vstupy sa odhaduje doba bežania procesu na 3-4 dni. Bude teda potrebné optimalizovať algoritmy KLFittera, aby bežal rýchlejšie

3.3 Požiadavky rozhrania:

Vybranie metódy: V rozhraní sa musí dať vybrať, či chceme robiť rekonštrukciu cez DNN alebo KLFitter

Heslo: Rozhranie bude zaheslované

Katalóg požiadaviek

Stav: Rozhranie bude zobrazovať, na koľko percent je proces dokončený

Katalóg požiadaviek

Prílohy:

- [1] ROOT: <https://root.cern.ch/>
- [2] <https://root.cern.ch/how/how-read-tree>
- [3] KLFitter: <https://github.com/KLFitter/KLFitter>
- [4] LWTNN: <https://github.com/lwttn/lwttn>